applied sciences

MDPI

Article

An Enhanced Deep Knowledge Tracing Model via Multiband Attention and Quantized Question Embedding

Jiazhen Xu * and Wanting Hu

Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China; huwanting@mails.ccnu.edu.cn

* Correspondence: xujiazhen@ccnu.edu.cn

Abstract: Knowledge tracing plays a crucial role in effectively representing learners' understanding and predicting their future learning progress. However, existing deep knowledge tracing methods, reliant on the forgetting model and Rasch model, often fail to account for the varying rates at which learners forget different knowledge concepts and the variations in question embedding covering the same concept. To address these limitations, this paper introduces an enhanced deep knowledge tracing model that combines the transformer network model with two innovative components. The first component is a multiband attention mechanism, which comprehensively summarizes a learner's past response history across various temporal scales. By computing attention weights using different decay rates, this mechanism adaptively captures both long-term and short-term interactions for different knowledge concepts. The second component utilizes a quantized question embedding module to effectively capture variations among questions addressing the same knowledge concept. This module represents these differences in a rich embedding space, avoiding overparameterization or overfitting issues. The proposed model is evaluated on popular benchmark datasets, demonstrating its superiority over existing knowledge tracing methods in accuracy. This enhancement holds potential for improving personalized learning systems by providing more precise insights into learners' progress.

Keywords: knowledge tracing; deep learning; transformer; self-attention; multiband attention; quantized question embedding

Citation: Xu, J.; Hu, W. An Enhanced Deep Knowledge Tracing Model via Multiband Attention and Quantized Question Embedding. *Appl. Sci.* **2024**, *14*, 3425. https://doi.org/10.3390/app14083425

Academic Editor: José Machado

Received: 18 March 2024
Revised: 13 April 2024
Accepted: 16 April 2024
Published: 18 April 2024

1. Introduction

Knowledge tracing (KT) is an innovative technique that has revolutionized the field of educational technology [1,2]. By leveraging data from learners' interactions with educational materials, knowledge tracing algorithms can accurately measure and predict their knowledge and skills over time. This approach enables educators and educational platforms to provide personalized and adaptive learning experiences for students, where the instruction is tailored to their individual needs and abilities. Through the analysis of patterns in learner responses, knowledge tracing algorithms can determine their current knowledge state, identify areas of strength and weakness, and make targeted recommendations for instruction. This personalized approach to learning not only enhances engagement and motivation but also leads to improved academic performance [3–6].

Inspired by the success of deep learning, recent studies on knowledge tracing have applied deep learning techniques. Deep knowledge tracing (DKT) [7] leverages recurrent neural networks (RNNs) [8] or long short-term memory neural networks (LSTMs) [9] to capture the temporal dependencies in learners' interactions. It takes into account the sequence of learners' responses along with additional factors such as the time spent on each task to predict their future performance accurately, and it has also demonstrated great potential for solving the knowledge tracing problem.

Several studies [10–14] have been conducted on transformer architecture, and researchers have explored the incorporation of attention mechanisms into knowledge tracing

models. These approaches address a drawback of the DKT model, which treats all questions within a sequence as equally important. While the specific attention mechanisms may differ, they share a common goal: to learn the attention weights for questions within a sequence of interactions. By doing so, these models aim to capture the relative importance of each question in predicting the likelihood of correctly answering the next question.

To better capture the dynamics of student knowledge and address the challenge of data sparsity in knowledge tracing tasks, existing transformer approaches essentially integrate two models based on prior knowledge: temporal decay attention and Rasch-model-based embedding.

Temporal decay attention: Psychological studies have shown that the consideration of forgetting is crucial for accurately estimating a student's knowledge state. This is because a student's mastery of knowledge tends to decline exponentially over time since their last practice with related questions [15]. Therefore, the standard attention function in a transformer is replaced by an attention function with a fixed-range temporal decay model [13,16].

Rasch-model-based embedding: In knowledge tracing tasks, the number of questions is often much larger than the number of knowledge concepts. To avoid data sparsity, and instead of directly embedding each question, an embedding representation that considers a parameter controlling the deviation of a question from the knowledge concept it involves is proposed by following the Rasch model [13,17,18].

Although the two models mentioned above can accurately characterize certain features of the learning process, we argue that there is still room for improvement in both of these models. First, because learners have different forgetting rates for various knowledge concepts, a fixed-range temporal decay attention model may not adequately accommodate these differences. Second, the Rasch model fixes the variation in the questions as a linear combination of two embedding vectors related to the knowledge concept covering them. Although it reduces the number of parameters, it also weakens the representative power of the model. Therefore, we propose an enhanced deep knowledge tracing model that operates via multiband attention and quantized question embedding. We summarize the main innovations of our work as follows:

- We introduce a multiband attention model that integrates multiple temporal scales to account for forgetting patterns at varying rates. This multiband model seamlessly aligns with the multihead attention mechanism in transformer models, wherein different heads employ distinct bands.
- To address the overfitting issue caused by having too many question tokens, we designed a quantized question embedding method to cluster the question embeddings during the training process, which effectively reduces the number of different embeddings. At the same time, this model does not impose limitations on its representation capacity in the embedding space.
- We conducted comprehensive experiments to assess the performance of our proposed model on four publicly available knowledge tracing datasets. The results clearly establish the effectiveness of our model.

The structure of this paper is outlined as follows: Section 2 provides a review of the literature related to our study. Section 3 enumerates the critical problem definitions and notations utilized in this research. Section 4 describes the knowledge tracing model we propose. Section 5 outlines the experimental design. Section 6 reports on the findings of our experiments and engages in discussions. Section 7 presents our conclusions.

2. Related Works

This section offers a snapshot of the existing research landscape in knowledge tracing.

2.1. Knowledge Tracing

Knowledge tracing [1] is a method used in educational technology to model and predict student knowledge and learning progress over time. It involves the analysis of

student data to infer their mastery of specific concepts or skills. The goal of knowledge tracing is to provide personalized feedback and support to students, thereby helping them to improve their learning outcomes.

Traditionally, knowledge tracing has been based on statistical models that use observed student responses to questions or tasks to estimate their current knowledge level. Bayesian knowledge tracing (BKT) [1,19] is a probabilistic modeling technique that uses Bayesian inference to estimate students' knowledge states. BKT assumes that students have a latent knowledge state that evolves over time and can be influenced by observables, such as their responses to learning tasks or assessment items. By capturing the probability of students mastering specific concepts, BKT helps in identifying their knowledge gaps and predicting their performance on future tasks. The Bayesian framework allows for updating the knowledge estimates as new information becomes available, thereby resulting in increasingly accurate predictions. The factor analysis model (FAM) [20,21] is a statistical technique that aims to identify the underlying latent factors responsible for student performance on learning tasks. It assumes that learners' knowledge and skills can be represented by a set of unobservable factors. FAM uses factor analysis to estimate the relationships between these factors and observable variables, such as student responses to assessment items or tasks. By capturing these latent factors, FAM helps in understanding the structure of students' knowledge and predicting their performance on future tasks.

Despite being a nascent field of research, knowledge tracing has already found diverse applications across various scenarios, such as learning resource recommendations and adaptive learning. Desmarais et al. [22] propose two extensions to the knowledge tracing model. The first extension takes into account the difficulty levels of exercises, while the second extension considers students' multiple attempts at solving exercises. These extensions are integrated into the knowledge tracing algorithm, which is used to recommend exercises based on students' knowledge states. Concretely, the algorithm begins by determining the expected range of scores for each exercise. It then calculates an expected score that the student should achieve in order to attain mastery, taking into account their current knowledge state. Then, the algorithm suggests the exercise with a predicted score closest to the expected score. As the knowledge state for a specific concept improves, the algorithm recommends more challenging exercises. Huang et al. [23] propose three additional objectives that are beneficial and specific for education. The first objective, "review and explore", aims to enhance students' understanding of nonmastered concepts through timely reviews and provide opportunities for exploring new knowledge. The second objective, "smoothness of difficulty level", suggests that the difficulty levels of consecutive exercises should be within a narrow range as students gradually learn new material. The third objective, "student engagement", focuses on recommending exercises that align with students' preferences, thereby promoting their enthusiasm during the learning process. To support online intelligent education with these specific objectives, the researchers developed a more advanced framework called multiobjective deep reinforcement learning. This framework incorporates three novel reward functions that capture and quantify the effects of the above objectives via knowledge tracing technology. Pardos et al. [19] implement knowledge tracing on the edX MOOC platform, specifically focusing on a 14-week online course that consists of weekly video lectures and accompanying lecture problems. The researchers utilize knowledge tracing to analyze students' learning behavior and improve their learning experience in the course. They accomplish this by creating effective learning strategies and dynamic learning paths that organize learning resources based on specific knowledge structures for the students. To maximize the utilization of cognitive structure for adaptive learning, Liu et al. [24] propose the Cognitive Structure Enhanced framework for adaptive Learning (CSEAL). CSEAL conceptualizes adaptive learning as a Markov decision process. The framework utilizes deep knowledge tracing to track the evolving knowledge states of students at each learning step. Furthermore, the authors develop a navigation algorithm that takes into account the knowledge structure, ensuring logical and reasonable learning paths in adaptive learning. This algorithm also reduces the search space

in the decision-making process. Finally, the CSEAL employs the actor–critic algorithm to dynamically determine the most appropriate learning resources for individual students, enabling sequential identification of the optimal resources for their learning progression.

2.2. Deep Knowledge Tracing

With the advancements in machine learning and natural language processing, more complex models have been developed, e.g., deep learning networks such as RNNs [8], LSTMs [9], and transformer-based [25] knowledge tracing frameworks.

Deep knowledge tracing (DKT) [7] has pioneered the use of deep learning for knowledge tracing. It uses RNNs and LSTMs to capture the sequential patterns in student responses and predict their mastery of concepts over time. DKT+ [10] introduces regularization terms for reconstruction and waviness into DKT's loss function in order to reconstruct observed inputs and keep consistency in predicting knowledge concept performance over time. DKT has been shown to outperform probabilistic and logistic models in terms of performance. However, it does have certain limitations that cannot be avoided. For example, it is challenging to understand how the hidden states can accurately represent student knowledge levels, and it cannot explicitly determine a student's level of knowledge mastery from the hidden state [26,27].

In order to enhance the interpretability of DKT, memory-aware knowledge tracing methods have been proposed. The dynamic key–value memory network (DKVMN) [11] approach uses a memory-augmented neural network to capture the relationship between student knowledge and learning materials. It maintains a memory matrix with key–value pairs, where the keys represent the exercise concepts and the values represent the students' knowledge levels. This method allows for more complex interactions between students and concepts, thus leading to better knowledge predictions. Abdelrahman et al. [28] highlighted a limitation of DKVMN, wherein they emphasized its inability to capture long-term dependencies in the learning process. To address this issue, they proposed a solution called the sequential key–value memory network (SKVMN). The SKVMN model combines the recurrent modeling capacity of DKT with the memory capacity of DKVMN, thereby aiming to overcome the shortcomings of both models.

2.3. Attentive Knowledge Tracing

The transformer model was initially introduced for neural machine translation [25]. Unlike traditional deep learning models that rely on recurrence, the transformer model completely relies on the self-attention mechanism to capture global dependencies within a sequence. This approach has demonstrated exceptional abilities in feature extraction and dependency capture while maintaining high computational efficiency. Several noteworthy pretraining models based on the transformer architecture, such as BERT [29] and GPT [30], have achieved state-of-the-art performance on various natural language processing tasks.

Pandey et al. [12] introduced a self-attentive model called SAKT (self-attentive knowledge tracing), which incorporates the transformer architecture to effectively capture the long-term dependencies in student learning interactions. This application of the transformer model directly contributes to achieving impressive performance in knowledge tracing. Wang et al. [31] introduced an adaptive sparse self-attention network, which not only generates the missing features but also provides detailed predictions of student performance. Zhu et al. [32] identified a vibration issue in DKT and proposed an attention-based KT model to address it. They also incorporated the use of finite state automaton for a comprehensive understanding of knowledge state transitions. Choi et al. [33] proposed an approach called separated self-attentive neural knowledge tracing (SAINT) to enhance the self-attentive computation for facilitating knowledge tracing adaptation. The SAINT model utilizes an encoder–decoder structure, where the embeddings of exercises and answers are independently encoded and decoded using self-attention layers. Shin et al. [34] introduced the SAINT+ model, which integrates two temporal features into SAINT. These features include the response time for each exercise and the duration between consecutive learning

interactions. Ghosh et al. [13] proposed a context-aware attentive knowledge tracing (AKT) model that combines the self-attention mechanism with psychometric models. The unique aspect of AKT is its implementation of a novel monotonic attention mechanism, which operates under the assumptions that the learning process is influenced by forgetting and that student knowledge tends to diminish over time. Wonsung Lee et al. [14] introduced a contrastive learning framework for knowledge tracing (CL4T) that focuses on revealing semantically similar or dissimilar examples of learning history to better understand their relationships.

Compared with other knowledge tracing methods, the attentive knowledge tracing or transformer framework has shown promising results in terms of predicting the future answers of learners.

Overall, knowledge tracing plays a crucial role in adaptive learning systems, offering insights into student progress and enabling personalized interventions. The transformer-based knowledge tracing framework, with its self-adaptive nature and enhanced modeling capabilities, has the potential to revolutionize the field of educational technology and improve student learning experiences.

3. Preliminaries

This section formalizes the research problem of knowledge tracing in this paper. For easy reference, Table A1 summarizes the symbols used throughout the paper.

3.1. Problem Definition

In knowledge tracing systems, a learner's learning activities consist of a sequence of questions and the corresponding responses. For learner i at time step t, they answer a question q_t drawn from a knowledge concept c_t and obtain response $r_t \in \{0, 1\}$, which denotes whether the learner correctly answers the question. Therefore, for each learner, we have their responses as a sequence

$$\{(q_1, c_1, r_1), \ldots, (q_t, c_t, r_t), \ldots, (q_t, c_T, r_T)\}, t \in 1, 2, \ldots, T, q_t \in P, c_t \in C, \tag{1}$$

where T is the length of the learning sequence, P is the set of all questions, and C is the set of all knowledge concepts.

Definition 1 (Knowledge Tracing). *Given the previous responses of a learner before time step t as a sequence $\{(q_1, c_1, r_1), \ldots, (q_t, c_t, r_t)\}$, as well as the question (q_{t+1}, c_{t+1}) at time step $t + 1$, the objective of knowledge tracing is to predict the response $\hat{r}_{t+1}$.*

3.2. Knowledge Concept Embedding

We tokenize and embed the knowledge concepts of the questions and answers in a similar way to Word2Vec [35]. Specifically, for exercise recording (q_t, c_t, r_t), c_t is converted into a one-hot vector $\tilde{c}_t$ of dimension $||C||$ based on the index of the knowledge concept present in the question. As for the answer to the question, in order to distinguish the different impacts on a learner's knowledge state between correct and incorrect answers, the response knowledge state is extended to a $2||C||$-dimensional one-hot vector $\tilde{r}_t$ as follows:

$$\tilde{r}_t = \begin{cases} \tilde{c}_t \oplus \mathbf{0} & \text{if } r_t = 1 \\ \mathbf{0} \oplus \tilde{c}_t & \text{if } r_t = 0, \end{cases} \tag{2}$$

where the feature vector $\mathbf{0} = (0, 0, \ldots, 0)$ has the same dimensions as $\tilde{c}_t$, and $\oplus$ denotes the concatenation operation.

Then, the one-hot vector $\tilde{c}_t$s and $\tilde{r}_t$s are passed through an embedding layer to convert them into embedding features for the knowledge concept of question and answer, respectively.

3.3. Temporal Decay Attention

Forgetting is a crucial factor in accurately assessing a student's knowledge, as their mastery of a subject tends to decline over time since their last practice [15]. The field of

attentive knowledge tracing has made efforts to model this forgetting effect. The most popular method is exponential temporal decay attention [13,16].

Let Q_t denote the query corresponding to the question the learner responds to at time step t, and let K_τ denote the key for the knowledge state at time step τ, then the standard attention function in the transformer network [25] is computed as follows:

$$a_{t,\tau} = \text{Softmax}\left(\frac{Q_t^T K_\tau}{\sqrt{D}}\right), \tag{3}$$

where D is the dimension of the query and key matrix.

The forgetting theory suggests that, when a learner encounters a new question, past experiences that are unrelated or from a long time ago are unlikely to be highly relevant. Therefore, the exponential decay attention is computed as follows:

$$\hat{a}_{t,\tau} = \text{Softmax}\left(\frac{Q_t^T K_\tau \cdot \exp(-\theta \cdot d(t,\tau))}{\sqrt{D}}\right), \tag{4}$$

where $\theta > 0$ is a decay rate parameter, and $d(t,\tau)$ is temporal distance measure between time steps t and τ.

3.4. Rasch-Model-Based Embedding

In real-world datasets, the number of knowledge concepts is often much smaller than the number of questions. To avoid overparameterization and overfitting when using deep learning methods to model the relationship between the questions and a learner's knowledge states, it is common to use the knowledge concepts included in the questions as tokens. Apparently, by doing this, the differences between different questions that contain the same knowledge concepts can be overlooked.

To this end, the Rasch model [13,18] has been widely applied to construct the variance of the questions. The Rasch model characterizes the probability that a learner answers a question correctly using two scalars: the question's difficulty and the learner's ability. Specifically, it constructs the embedding of the question q_t from knowledge concept c_t at time step t as

$$x_t = e_{c_t} + \mu_{q_t} \cdot d_{c_t}, \tag{5}$$

where e_{c_t} is the embedding of the concept this question covers, d_{c_t} is a vector that summarizes the variation in questions covering this concept, and μ_{q_t} is a scalar difficulty parameter that controls how far this question deviates from the knowledge concept it covers. Since the embedding vectors e_{c_t} and d_{c_t} are entirely based on the knowledge concept c_t, and as the scalar μ_{q_t} is based on the question q_t, it can greatly reduce the number of parameters so as to avoid overfitting.

4. Methodology

In this section, we introduce our proposed model in detail, which mainly consists of two novel modules: the quantized question embedding module and the multiband attention module.

4.1. Overall Architecture

The overall architecture of the adaptive transformer is shown in Figure 1.

The proposed network primarily consists of n layers of stacked encoder and decoder [25]. The question data go through knowledge concept embedding (KC embedding) and quantized question embedding (QQE), respectively. The results are summed and fed into the encoder, which is composed of multiband attention (MBA) and feed-forward layers, to learn the interactions between the questions. The output is then combined with the embedding of the response data and fed into the decoder. The decoder, which is composed of masked multiband attention and feed-forward layers, learns the relationship between the question and the learner's knowledge state. The output of the decoder is passed through

linear and Softmax layers to obtain the learner's response. The final loss of the network is a weighted sum binary cross entropy loss of the prediction and mean square loss of the features of the quantized question embedding codebook. We provide a detailed explanation of these modules later.

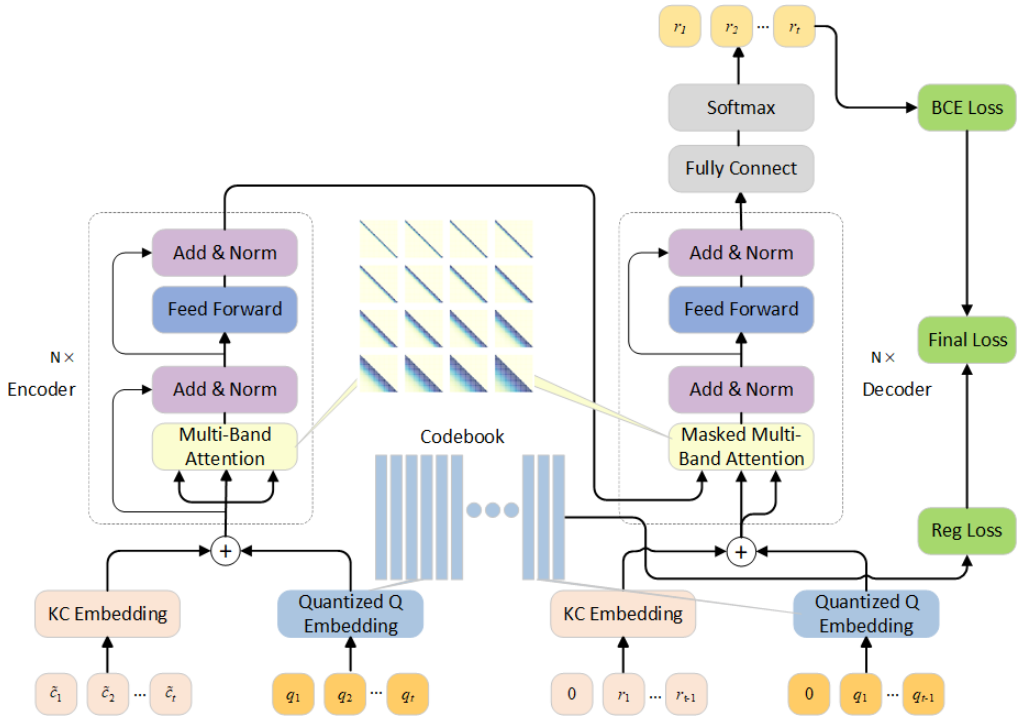


Figure 1. Overall architecture of the proposed network. The model consists of stacked encoder and decoder layers. It incorporates the knowledge concept and quantized question embeddings for question data, which are then processed through multiband attention and feed-forward layers in the encoder to capture question interactions. The resulting output is combined with response data and fed into the decoder, which learns the relationship between the question and knowledge state. The decoder's output undergoes linear and Softmax layers to predict the learner's response.

4.2. Quantized Question Embedding

Knowledge concepts describe the main characteristics of a question, but regarding them as completely equivalent overlooks the subtle variation between different questions that share the same knowledge concept. In real-world data, the number of questions is often much larger than the number of knowledge concepts, thereby resulting in a data sparsity problem—in other words, directly tokenizing and embedding the questions can lead to overparameterization and overfitting issues.

To address this, we employed quantized question embedding to tackle this problem. As shown in Figure 2, a codebook was created to store the quantized question embedding features, where the number of features in the codebook was much smaller than the number of questions. For each question q_t, we first used an embedding layer to obtain its embedding feature z_t, Then, the model searches for the most similar feature x_t in the codebook and sums it up with the knowledge concept embedding, which serves as the input to the encoder and decoder. Intuitively, the features stored in the codebook can be seen as several cluster centers of the questions' variations. Instead of directly using the embedding features of the questions, these quantized cluster center features help to avoid overparameterization and overfitting.

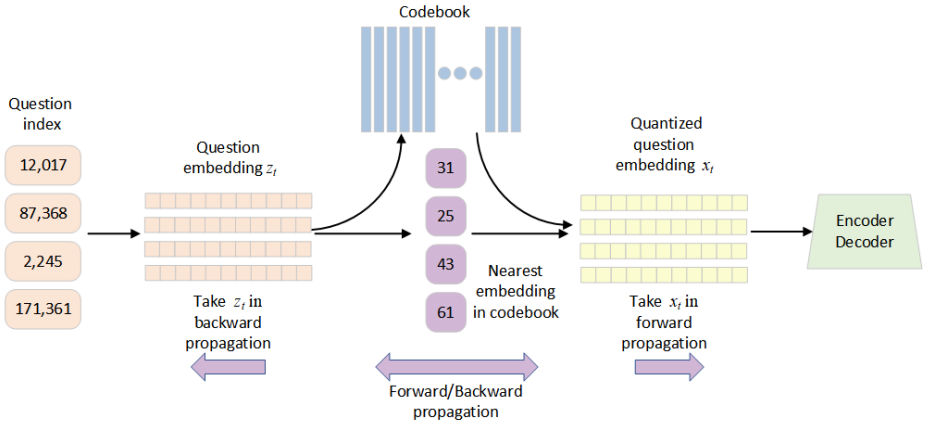


Figure 2. The process of quantized question embedding. We implemented quantized question embedding by creating a codebook to store quantized features. For each question embedding z_t, the model searched for the most similar feature x_t in the codebook. A stop gradient operation was introduced to solve the nondifferentiable process from z_t to x_t. For this module, during forward propagation, we replaced z_t with x_t. During backward propagation, we replaced x_t with z_t.

The transfer process from z_t to x_t is nondifferentiable. Consequentially, we introduced the stop gradient [36] operation as follows:

$$sg(x) = \begin{cases} x & (in\ forward\ propagation) \\ 0 & (in\ backward\ propagation). \end{cases} \tag{6}$$

During forward propagation, the value inside the stop gradient operation remains unchanged. During backward propagation, the gradient of the stop gradient operation is set to zero, indicating that this calculation has no gradient. With this operation, we can incorporate the output of the quantized question embedding module into the network as follows:

$$Ouput_{QQE}(x_t, z_t) = z_t + sg(x_t - z_t), \tag{7}$$

where it forward propagates x_t to the next layer of the framework and backpropagates the gradient of z_t to the previous layer.

4.3. Multiband Attention Model

The forgetting mechanism effectively constrains interactions within a relatively narrow range, thereby allowing the current knowledge concept to consider more recent exercise results. However, due to the varying forgetting rates for different knowledge concepts among learners, a fixed-parameter forgetting model may not necessarily fit it well. Therefore, we propose a multiband attention model, which possesses multiple effective temporal ranges of interaction to accommodate forgetting patterns with different rates. This multiband attention model naturally adapts to the multihead attention mechanism of transformer models, where different heads utilize different bands.

The original input is, respectively, processed through a knowledge concept embedding layer and quantized question embedding layer, and it is then summed up to obtain the question representation feature $\hat{x}_t$ and the response representation feature $\hat{y}_t$. Next, we used a multiband attention model and masked multiband attention layers to capture the interactions between features.

We obtained the query, key, and value for the question embeddings $\{\hat{x}_1, \ldots, \hat{x}_t\}$ with the following equations:

$$Q = \hat{X}W_q, K = \hat{X}W_k, V = \hat{X}W_v, \tag{8}$$

where $\hat{X}$ is the question feature matrix that is composed of vectors $\{\hat{x}_1, \ldots, \hat{x}_t\}$, and W^q, W^k, and W^v are the linear projection matrices for query, key, and value, respectively.

The relevance of each of the previous interactions with the current question is determined by the mth band attention weights, which are defined as follows:

$$Attention_m(Q, K, V) = \text{Softmax}\left(\frac{QK^T exp(-m\theta d(\Delta t))}{\sqrt{D}}\right)V, \tag{9}$$

where $d(\Delta t)$ denotes the distance between the temporal interval Δt, and the parameters θ control the decay rate of the distance.

To effectively attend to the information from various representative subspaces, we utilized the projection matrices W_qs, W_ks, and W_vs to linearly project the queries, keys, and values with different ms. Figure 3 illustrates the combination of multiband attention matrices with different ms from 1 to 8.

$$Multiband(x) = (band_1 \oplus \ldots \oplus band_h)W_o, \tag{10}$$

where $band_m = Attention_m(\hat{X}W_q^{(m)}, \hat{X}W_k^{(m)}, \hat{X}W_v^{(m)})$, and W_o is a linear projection matrix.

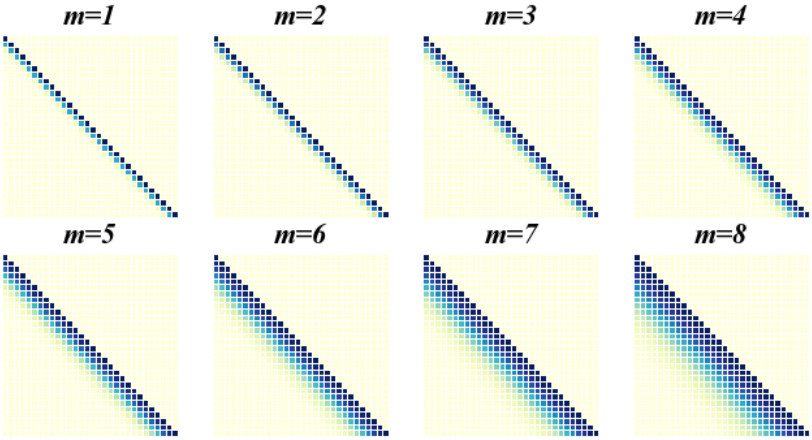


Figure 3. The combination of multiband attention matrices with different ms. Blue areas represent the effective attention weights that can be obtained. The greater the depth of blue, the less the degree of forgetting. Different bands captured the attention information within different time step ranges.

Similarly, the query, key, and value for the response embeddings $\{0, \hat{y}_1, \ldots, \hat{y}_{t-1}\}$ were expressed with the following equations:

$$Q = \hat{Y}\tilde{W}_q, K = \hat{Y}\tilde{W}_k, V = \hat{X}\tilde{W}_v, \tag{11}$$

where $\hat{Y}$ is the response feature matrix composed of vectors $\{0, \hat{y}_1, \ldots, \hat{y}_{t-1}\}$.

The masked attention operation was adopted to prevent the model from seeing future answers.

$$MaskedAttention_m(Q, K, V) = \text{Softmax}\left(\frac{QK^T exp(-m\theta d(\Delta t)) \odot M}{\sqrt{D}}\right)V, \tag{12}$$

where M is a lower triangular unit matrix, and $\odot$ denotes element-wise multiplication.

4.4. Encoder and Decoder

The main components of the proposed network were the n encoders and decoders. The encoders were responsible for learning the interactions between the different questions and converting input question embeddings into intermediate representations, which can be seen as the question's knowledge requirement. Meanwhile, the decoder learns the relationship between the intermediate question representations and response embeddings, as well as converting the response embeddings into high-level representations, which can be seen as a learner's knowledge state.

As shown in Figure 1, both the encoder and decoder adopted a similar residual connection network structure. For the encoder, the input question embedding serves as the query, key, and value. It goes through a multiband attention layer, a layer-normalization layer, a feed-forward layer, and another layer-normalization layer to obtain the intermediate representations of the questions. For the decoder, it takes the response embedding as the query and key, and the output of the encoder is taken as the value. It goes through a masked multiband attention layer, a layer-normalization layer, a feed-forward layer, and another layer-normalization layer to obtain the high-level representations of the answers. Both the encoder and decoder are stacked with n layers, where the output of the previous layer serves as the input to the next layer.

The output of the last decoder obtained above is passed through the fully connected layer with Sigmoid activation to predict the response of the learner.

$$s_t = \text{Sigmoid}(w_f^T d_t + b_f), \tag{13}$$

where s_t is a scalar representing the probability of the learner providing the correct response to question t, and w_f and b_f are the parameters of the fully connected layer.

The main objective of the training was to minimize the negative log-likelihood of the observed sequence of learner responses under the model. The parameters were learned by minimizing the binary cross entropy (BCE) loss between s_t and r_t.

$$L_{BCE} = -\sum_t r_t log(s_t) + (1 - r_t) log(1 - s_t). \tag{14}$$

As for regularization, we hope that the embedding features of all the questions related to the same knowledge concept were close to the embedding of the knowledge concept itself. In other words, the sum of the norms of all the features in the codebook, i.e.,

$$L_{Reg} = \sum_{i \in B} \|x_i\|^2, \tag{15}$$

where x_is are all the embedding features in codebook B, which should be as small as possible.

The final loss function was a weighted sum of L_{BCE} and L_{Reg}.

$$L = \lambda L_{BCE} + (1 - \lambda) L_{Reg}, \tag{16}$$

where λ is the hyperparameter to govern the influence of question embedding variance.

5. Experiments

In this section, we present the experimental settings aimed at evaluating our proposed model. This evaluation is conducted by addressing the following crucial research questions:

- Does our proposed model outperform other knowledge transfer models?
- How do the multiband attention and quantized question embedding components within our model contribute to its overall performance?
- What is the explanation for the impact of these components on the model's effectiveness?

5.1. Datasets

We conducted experiments on four popular datasets to evaluate the effectiveness of the proposed transformer in various learning scenarios. Table 1 presents the statistical details of these datasets. We briefly introduce these datasets as follows:

- Algebra05 (https://www.kdd.org/kdd-cup/view/kdd-cup-2010-student-performance-evaluation/Data, accessed on 10 March 2024) was introduced during the KDDcup 2010 Educational Data Mining challenge [37]. It comprises student responses to algebra questions from 2005 to 2006.
- Bridge06 (https://www.kdd.org/kdd-cup/view/kdd-cup-2010-student-performance-evaluation/Data, accessed on 10 March 2024) is similar to Algebra05, but it includes more learners, questions, and interactions from 2006 to 2007 [37].

- Assist09 (https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data, accessed on 10 March 2024) is a dataset collected from ASSISTment, an online tutoring system for grade school math exercises, which contains student responses to different types of questions, such as multiple choice, text, and open-ended questions [38].
- Assist17 (https://sites.google.com/view/assistmentsdatamining/dataset, accessed on 10 March, 2024) is another collection from ASSISTment for a data mining competition in 2017, with learners having significantly longer learning sequences [39]. This dataset allows learners to attempt a single question multiple times until they reach the correct answer. Therefore, it becomes crucial for models to take into account the cumulative effect throughout the learning process.

Table 1. The statistics of the datasets.

Dataset	Learners	Questions	Skills	Interactions
Algebra05	571	173,113	271	607,014
Bridge06	1138	129,263	550	1,817,450
Assist09	4163	17,751	123	338,001
Assist17	1709	3162	102	942,816

5.2. Evaluation Metrics

The goal of knowledge tracing is to predict learner performances by combining classification and regression methods. This research utilizes two evaluation metrics to measure the accuracy of the prediction models: the area under the curve (AUC) and the root mean square error (RMSE). These metrics provide a comprehensive assessment of the models' effectiveness in predicting learner outcomes.

The AUC is a commonly used evaluation metric in binary classification tasks. In knowledge tracing research, the AUC metric is utilized to assess the ability of prediction models to correctly classify learner outcomes and discriminate between correct and incorrect responses. Higher AUC values indicate a more accurate and reliable model in predicting learner performance.

The RMSE is a commonly used evaluation metric in regression tasks. It measures the average deviation between the predicted values of a model and the actual values. In the context of knowledge tracing, a smaller RMSE value signifies improved prediction performance, as values closer to 0 indicate a higher accuracy in predictions.

5.3. Baseline Methods

To evaluate the overall performance of the proposed model, we conducted a comparison between our method and the various established KT methods. In order to ensure a fair comparison, we utilized the optimal parameter settings for each method. An overview of the baseline methods is provided below:

- DKT [7] was the first approach to incorporate deep learning techniques into knowledge tracing. It employs either an RNN or LSTM to represent the knowledge state as a multidimensional hidden state during the learning process.
- DKT+ [10] enhances the DKT method by introducing regularization terms into its loss function. These terms focus on reconstruction and waviness to ensure an accurate reconstruction of observed inputs, and they maintain consistency in predicting the performance of knowledge concepts over time.
- DKVMN [11] enhances the hidden variable representation of DKT by incorporating a memory network. This memory structure comprises two matrices: a key matrix to store all the concepts in a static manner and a value matrix for the dynamic storage and retrieval of the mastery levels of each concept through reading and writing operations.
- SAKT [12] is the initial knowledge tracing model that utilizes the transformer architecture with an attention mechanism. This attention mechanism evaluates the

significance of prior questions in relation to the entire learning sequence, thereby enabling the prediction of learning performances with respect to the current question.
- AKT [13] employs a context-aware attention mechanism to acquire context-aware representations of exercises and answers. Unlike SAKT's scaled dot-product attention, AKT introduces a modified monotonic attention approach to mimic the forgetting effect through the exponential decay of attention weights.
- CL4KT [14] presents a framework for knowledge tracing that utilizes contrastive learning to highlight the similarities or differences between examples of learning history in order to gain deeper insights into their relationships.

5.4. Ablation Study

We conducted an ablation study in order to examine the influence of different components on the overall performance prediction in DCKT. This study included numerous variants of our proposed model, which allowed us to assess the impact of the multiband attention module and the quantized question embedding module.

5.5. Visualization

To gain a better understanding of the effects of the multiband attention and quantized question embedding modules, as well as the overall performance of the framework, we conducted an experiment to visualize the output of the real data processed through these modules and the framework.

For the MBA module, we compared the attention maps generated using the MBA module with the ones generated directly from the temporal distance decay methods. We then observed the differences between these two approaches.

Regarding the QQE module, we obtained embedding features for knowledge concepts and questions with and without QQE. We then employed the t-SNE [40] method to map these features into three-dimensional space and observed the differences in their distributions. The t-SNE (t-distributed stochastic neighbor embedding) approach is a dimensionality reduction technique commonly used for visualizing high-dimensional data in a lower-dimensional space. It aims to preserve the local relationships between data points while simultaneously spreading them out to reveal global patterns.

To evaluate the overall performance of the framework, we randomly selected 10, 20, 50, and 100 concepts from the dataset and calculated the embedding features for all the questions associated with these concepts. We then mapped these features onto a two-dimensional space using t-SNE to observe whether they were adequately separated.

5.6. Experimental Setup and Implementation Details

We followed the standard detailed in [41] to preprocess the datasets. We filtered out learners with less than five interactions and removed interactions not associated with the named concepts. For quantitative evaluation, we employed a five-fold cross-validation approach, with the folds being based on individual learners. We implemented the proposed model in PyTorch 2.0 [42] with an NVIDIA (Santa Clara, CA, USA) GeForce RTX 2080Ti GPU card. The models were optimized by the Adam optimizer [43] with a batch size of 64 and an initial learning rate of 0.001. The layers of the encoder and decoder are both set to 3. The size of the embedding and hidden features was set to 256, and the number of bands was set to 8. The number of the QQE codebook was set to 1024. The regularization parameter λ was set to 0.01.

6. Results and Discussion

In this section, we present the experimental results and discuss the findings from our experiments.

6.1. Overall Performance

To evaluate the overall performance of our model, we compared our model with the state-of-the-art knowledge tracing models, such as DKT, DKT+, DKVMN, SAKT, AKT, and CL4KT, on all four of the datasets mentioned above. As shown in Table 2, we compared our model with all the baseline methods under RMSE and AUC. Figures 4 and 5 visualize the RMSE and AUC values, respectively, with bar plots. They show that our model outperformed the other models in both metrics, especially Assist17. In the Assist17 dataset, the method with the best performance apart from ours is the CL4KT method. The RMSE and AUC values of our method are 37.36% and 85.75%, respectively, while the CL4KT method has RMSE and AUC values of 42.27% and 78.06%. Compared with the CL4KT method, our method improved the RMSE and AUC values by 4.9% and 7.7%, respectively. Although Assist17 has fewer learners and questions compared with Assist09, the average number of interactions per learner is significantly higher in Assist17 than in Assist09. Our approach achieved a more significant performance on Assist17, which demonstrates that our model excels at learning patterns from long-term sequence data and that it can also adapt to short-term sequence data. Additionally, the proposed model shows only marginal improvements on Algebra05 and Bridge06 datasets, primarily due to the intricate nature of the latent knowledge structure within the datasets. Both datasets entail a significantly higher number of questions, posing a significant challenge for knowledge tracing.

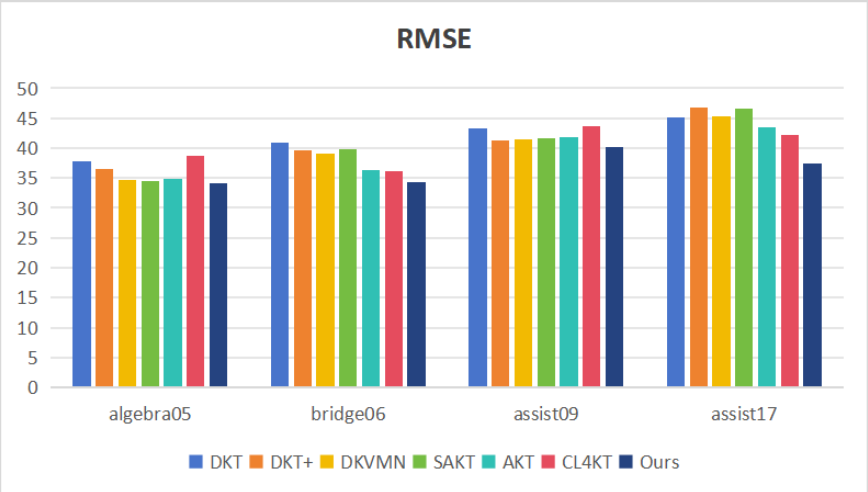


Figure 4. The RMSE values of all the KT methods over four datasets.

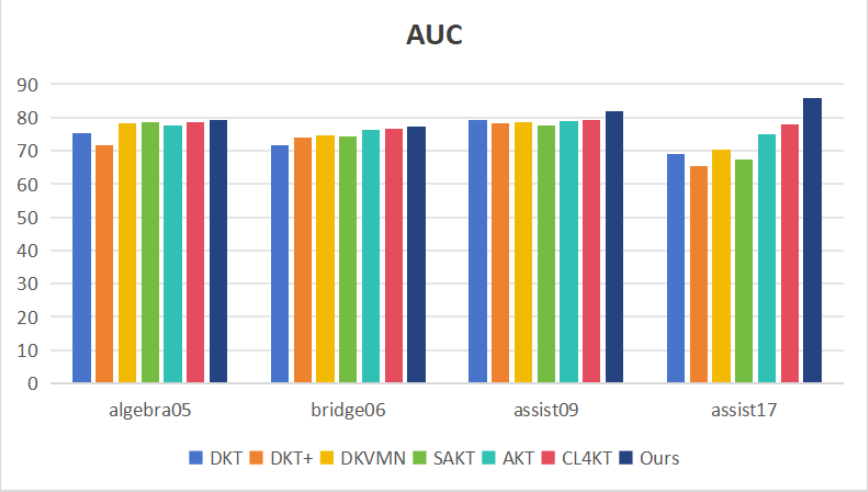


Figure 5. The AUC values of all the KT methods over four datasets.

Table 2. The overall performance comparison of the four datasets. Our model outperformed other baseline models (especially the assist17 dataset).

Dataset	Metric	DKT	DKT+	DKVMN	SAKT	AKT	CL4KT	Ours
Algebra05	RMSE	37.85	36.42	34.63	34.57	34.91	38.65	34.12
	AUC	75.14	71.63	78.37	78.48	77.61	78.49	79.26
Bridge06	RMSE	40.91	39.56	39.01	39.77	36.26	36.10	34.27
	AUC	71.82	74.07	74.65	74.43	76.18	76.65	77.41
Assist09	RMSE	43.27	41.19	41.53	41.71	41.86	41.59	40.25
	AUC	79.25	78.16	78.64	77.52	78.81	79.30	81.82
Assist17	RMSE	45.06	46.71	45.38	46.62	43.48	42.27	37.36
	AUC	68.98	65.37	70.29	67.21	74.95	78.06	85.75

6.2. Ablation Studies

In this section, we present the ablation studies for the proposed model.

To verify the effects of the multiband attention and quantized question embedding, we conducted comparative experiments with and without them on the Assist09 and Assist17 datasets. As shown in Table 3, the network equipped with multiband attention and quantized question embedding outperformed the one without them, thereby indicating the effectiveness of the proposed embedding modules.

Table 3. The performance comparison of our framework with and without MBA, as well as of the QQE modules on the Assist09 and Assist17 datasets.

Dataset	Metric	Baseline	+MBA	+MBA + QQE
Assist09	RMSE	42.84	41.13	40.25
	AUC	78.75	80.58	81.82
Assist17	RMSE	43.62	41.15	37.36
	AUC	74.80	78.26	85.75

We conducted experiments to optimize the hyperparameters of the proposed model, such as the number of the bands and the size of the codebook, on the Assist09 and Assist17 datasets. Using various hyperparameters, we trained and tested multiple neural network models. For each model, we evaluated its performance using RMSE and AUC. Table 4 demonstrates the impact of selecting varying numbers of bands in the multiband attention module on the final performance, while keeping all other settings the same. Table 5 exhibits the effect of choosing different codebook sizes in the quantized question embedding module on the overall performance, while maintaining consistent settings in all other aspects. From Tables 4 and 5, it can be observed that different hyperparameter selections have minimal impact on the overall performance of the model, indicating that our model is relatively stable. The hyperparameters yielding optimal performance are the band set to 8 and the codebook size set to 1024.

Table 4. The performance comparison of the proposed model with 4, 8, 16, and 32 bands on the Assist09 and Assist17 datasets.

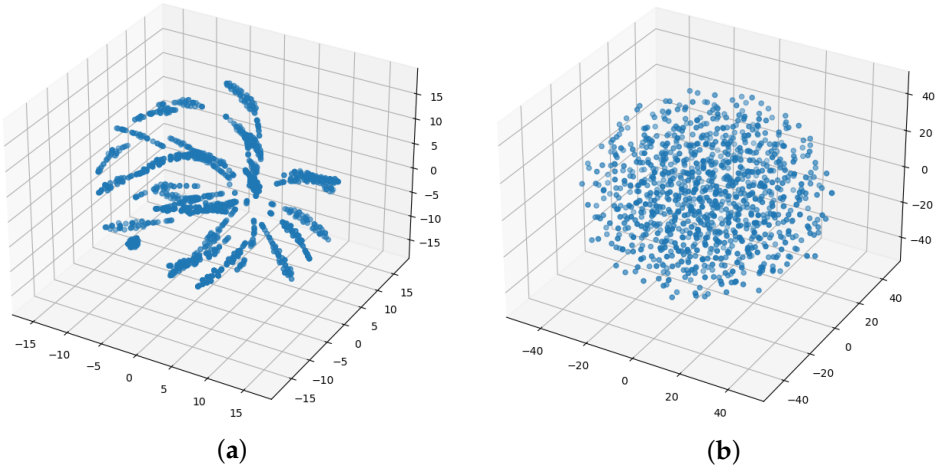
Dataset	Metric	4	8	16	32
Assist09	RMSE	40.71	40.25	40.32	40.98
	AUC	81.26	81.82	81.63	80.91
Assist17	RMSE	37.79	37.36	37.44	37.96
	AUC	85.21	85.75	85.62	85.03

Table 5. The performance comparison of the proposed model with 256, 512, 1024, and 2048 codebook sizes on the Assist09 and Assist17 datasets.

Dataset	Metric	256	512	1024	2048
Assist09	RMSE	40.91	40.37	40.25	40.32
	AUC	81.33	81.64	81.82	81.76
Assist17	RMSE	37.92	37.68	37.36	37.44
	AUC	84.91	85.32	85.75	85.61

6.3. Visualization

Figure 6 shows the distribution of the questions belonging to a knowledge concept in the embedding space. For the sake of visualization, we used the t-SNE technique to map the original space to a 3D space. Figure 6a represents the Rasch model, while Figure 6b represents our QQE model.



(**a**) (**b**)

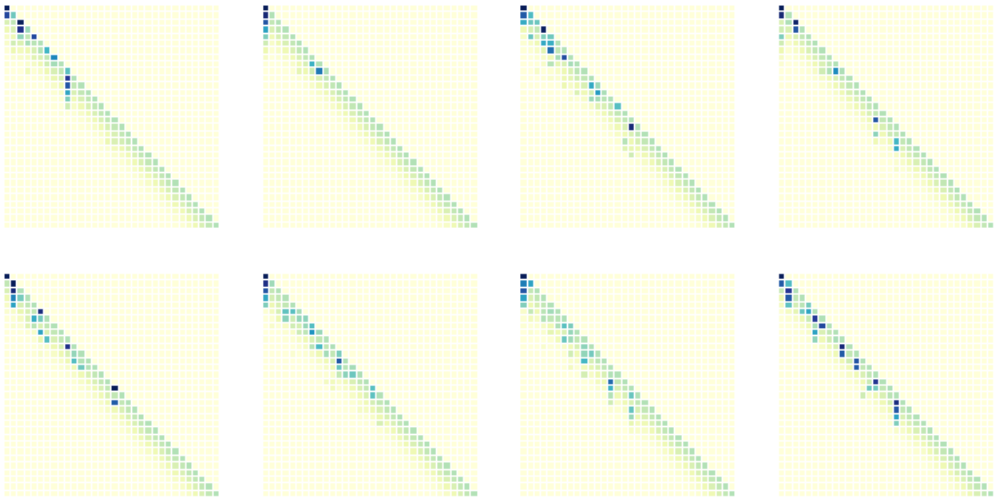
Figure 6. The distribution of a knowledge concept's questions in the embedding space: (**a**) The questions of a knowledge concept in the Rasch model. (**b**) The questions of a knowledge concept in the QQE model.

The deep knowledge tracing algorithms based on the Rasch model [13,14,16] assume that all the questions belonging to a knowledge concept can be expressed as a linear combination of two vectors, which are entirely determined by the knowledge concept. Therefore, in the original embedding space, the data points representing these questions are distributed along a line segment. When mapped to the 3D space, they are squeezed into a particularly narrow manifold. On the other hand, after learning with the QQE model, the embedding points of the questions were distributed in a spherical space that resembled a Gaussian distribution. This distribution can more naturally reflect the relationship between the questions and the knowledge concept. This difference shows that our model alleviates the overparameterization problem without introducing overly strong prior assumptions, resulting in better representations and better predictions.

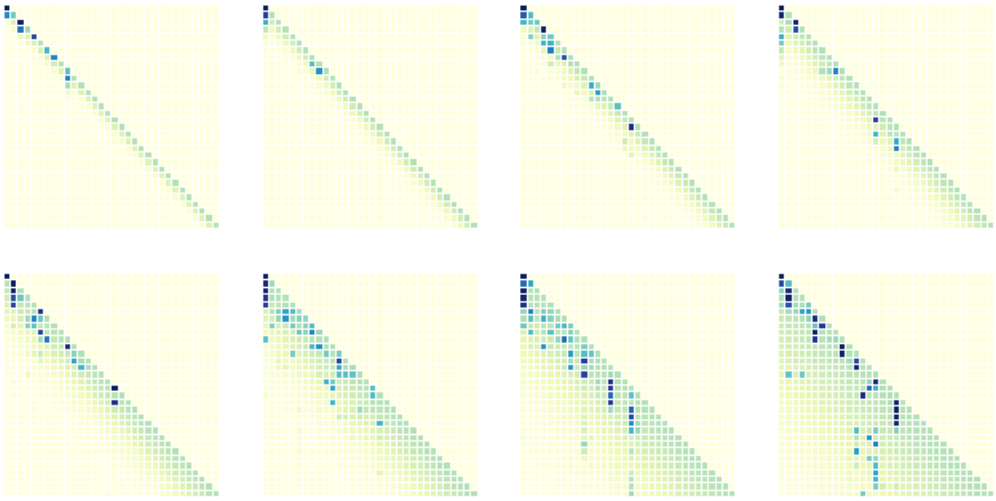
Figure 7 illustrates the hotmaps of using and not using the multiband attention obtained from the query and key multiplication. Both Figure 7a and Figure 7b utilize eight individual heads. In Figure 7a, the eight heads adopt the same temporal decay, while in Figure 7b, each of the eight heads employs different temporal decay.

It can be observed that different bands can capture attention within different time interval ranges, which enables a better adaptive fusion of features from different bands. Whether or not the multiband strategy is used, the attention model requires the multihead attention mechanism. Each head employing a different band will not increase computational overhead, thus leaving the computational complexity unchanged.

Figure 8 shows the distribution of the knowledge concepts and questions on the Assist09 dataset in the embedding space, which are mapped to a two-dimensional space using t-SNE. Each dot represents a question, and each plus sign represents a concept, with different colors distinguishing the questions belonging to different concepts. Figure 8a–d shows the distribution of 10 knowledge concepts, 20 knowledge concepts, 50 knowledge concepts, and 100 knowledge concepts that were randomly selected, as well as their associated questions, respectively. It indicates that our model has successfully extracted meaningful, discriminative embedding features from the raw one-hot encoding vectors. These features capture essential differences between different knowledge concepts such that the questions containing the same knowledge concept are clustered together in the low-dimensional embedding space, while distinct knowledge concepts maintain appreciable distances from one another. Even in the reduced-dimensional view, the model is able to effectively identify and distinguish among knowledge concepts. The good separation of the questions and knowledge concepts also indicates that the model has learned embeddings that preserve the inherent knowledge structure of the data.



(**a**)



(**b**)

Figure 7. The hotmaps of the different heads' attention weight matrices: (**a**) The attention weight matrix where the heads adopt the same temporal decay. (**b**) The attention weight matrix of the multiband attention model.

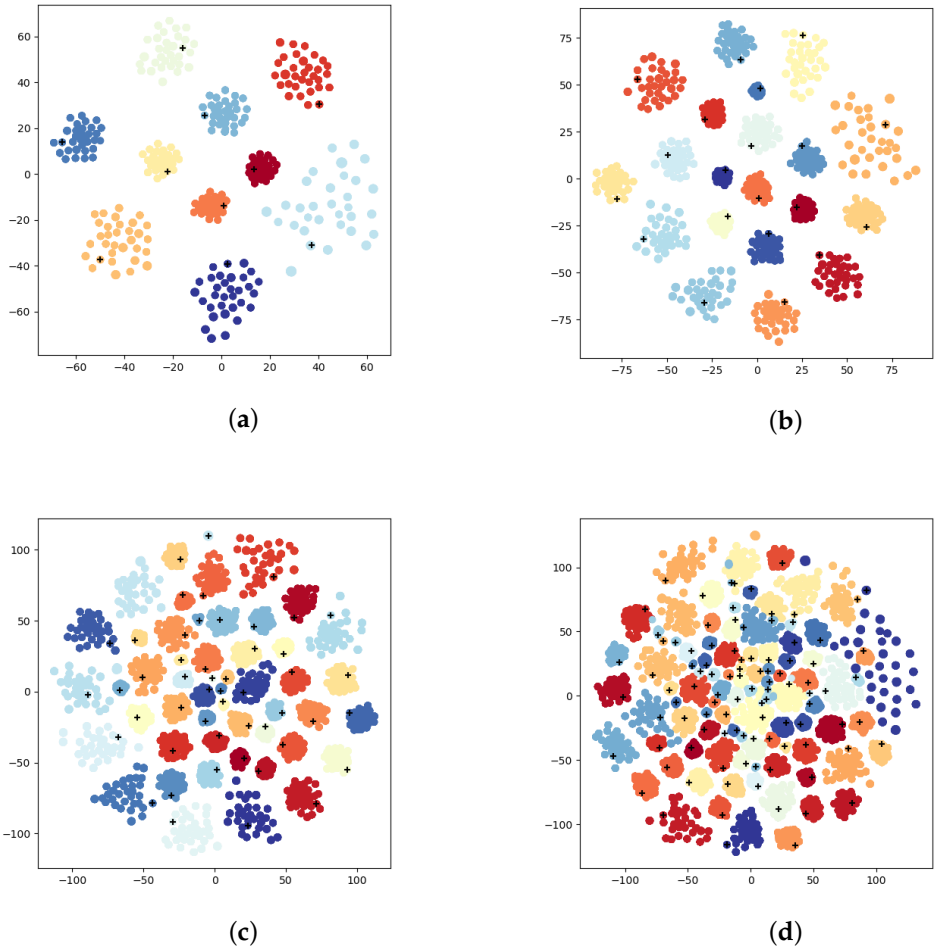


Figure 8. The distribution of 10 knowledge concepts (**a**), 20 knowledge concepts (**b**), 50 knowledge concepts (**c**), and 100 knowledge concepts (**d**) that were randomly selected, as well as their associated questions on the Assist09 dataset, respectively.

7. Conclusions

In this paper, we introduced an enhanced deep knowledge tracing method that leverages transformer networks. The proposed model incorporates knowledge concepts and quantized question embeddings for question data, which are then processed through multiband attention and feed-forward layers in the encoder to capture question interactions. The resulting output is combined with response data and fed into the decoder, which learns the relationship between the question and knowledge state. This method enhances existing knowledge tracing techniques through the employment of a multiband attention mechanism. This mechanism effectively summarizes the past response history of learners across various temporal scales. Different bands capture attention within different time interval ranges, which enables a better adaptive fusion of features. Additionally, we utilized quantized question embedding to adaptively capture the subtleties among the questions addressing the same knowledge concept. Compared with the Rasch model, the distribution of the question embedding more naturally reflects the relationship between the questions and the knowledge concept. Our experimental findings on the benchmark datasets revealed that our method surpasses the state-of-the-art knowledge tracing methods.

Despite the promising results achieved by the proposed models, limitations still exist that need to be addressed. As an illustration, the model requires complete annotations of the knowledge concepts contained in all questions. Moreover, the current model relies on abstract format data, consisting of one-hot encoding of questions, knowledge concepts, and learners' responses, without taking into account their textual content. With the development

of large language models, further exploration of the connections within the textual content itself is warranted.

In our future research, we plan to explore various opportunities for better utilization of multimodal information. One avenue we will explore is the integration of textual content to enhance the embedding representations for questions, knowledge concepts, and responses. We will also explore automatically annotating knowledge concepts for questions through this textual content, thereby reducing the workload of manual annotation and obtaining more training data.

Author Contributions: Conceptualization, J.X.; methodology, J.X.; validation, W.H.; formal analysis, J.X. and W.H.; investigation, J.X.; resources, J.X.; writing—original draft preparation, J.X.; review and editing, W.H.; visualization, J.X.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the MOE Project of Humanities and Social Sciences under grant No. 18YJC880096 and the National Natural Science Foundation of China under grant No. 62277029.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. A list of important notations.

Variable	Description
q_t	The question at time step t
c_t	The knowledge concept in q_t
r_t	The learner's response to q_t
$\tilde{c}_t$	The one-hot vector for the knowledge concept embedding representation in question q_t
$\tilde{r}_t$	The one-hot vector for the knowledge concept embedding representation in response r_t
$\tilde{q}_t$	The one-hot vector for q_t embedding representation
z_t	The embedding representation of q_t
x_t	The quantized embedding representation of q_t
$\hat{x}_t$	The fused embedding representation of q_t and c_t
$\hat{y}_t$	The embedding representation of r_t
$\hat{X}$	The matrix composed of $\{\hat{x}_1, \ldots, \hat{x}_t\}$
$\hat{Y}$	The matrix composed of $\{\hat{y}_1, \ldots, \hat{y}_t\}$
Q	The query matrix in attention function
K	The key matrix in attention function
V	The value matrix in attention function

References

1. Corbett, A.; Anderson, J. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User Adapt. Interact.* **1994**, *4*, 253–278. [CrossRef]
2. Liu, Q.; Shen, S.; Huang, Z.; Chen, E. A Survey of Knowledge Tracing. *arXiv* **2021**, arXiv:2105.15106.
3. Chrysafadi, K.; Virvou, M. Student modeling approaches: A literature review for the last decade. *Expert Syst. Appl.* **2013**, *40*, 4715–4729. [CrossRef]
4. Nguyen, T. The efectiveness of online learning: Beyond no signifcant diference and future horizons. *Merlot J. Online Learn. Teaching.* **2015**, *11*, 309–319.
5. Mohammadi, H. Investigating users' perspectives on e-learning: An integration of TAM and IS success model. *Comput. Hum. Behav.* **2015**, *45*, 359–374. [CrossRef]

6. Wang, F.; Liu, Q.; Chen, E.; Huang, Z.; Chen, Y.; Yin, Y.; Huang, Z.; Wang, S. Neural cognitive diagnosis for intelligent education systems. In Proceedings of the AAAI Conference on Artifcial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 6153–6161.
7. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.; Sohl-Dickstein, J. Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 505–513.
8. Williams, R.; Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1989**, *1*, 270–280. [CrossRef]
9. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
10. Yeung, C.; Yeung, D. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale, London, UK, 26–28 June 2018; pp. 1–10.
11. Zhang, J.; Shi, X.; King, I.; Yeung, D. Dynamic key-value memory networks for knowledge tracing. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 765–774.
12. Pandey, S.; Karypis, G. A self-attentive model for knowledge tracing. In Proceedings of the 12th International Conference on Educational Data Mining, Montreal, QC, Canada, 2–5 July 2019; pp. 384–389.
13. Ghosh, A.; Hefernan, N.; Lan, A. Context-aware attentive knowledge tracing. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual, CA, USA, 23–27 August 2020; pp. 2330–2339.
14. Lee, W.; Chun, J.; Lee, Y.; Park, K.; Park, S. Contrastive Learning for Knowledge Tracing. In Proceedings of the ACM Web Conference, Lisbon, Portugal, 21–24 May 2022; pp. 2330–2338.
15. Pashler, H.; Cepeda, N.; Lindsey, R.; Vul, E.; Mozer, M. Predicting the optimal spacing of study: A multiscale context model of memory. In Proceedings of the Conference on Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 1321–1329.
16. Nagatani, K.; Zhang, Q.; Sato, M.; Chen, Y.Y.; Chen, F.; Ohkuma, T. Augmenting knowledge tracing by considering forgetting behavior. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3101–3107.
17. Lord, F. *Applications of Item Response Theory to Practical Testing Problems*; Erlbaum Associates: Mahwah, NJ, USA, 1980.
18. Rasch, G. *Probabilistic Models for Some Intelligence and Attainment Tests*; MESA Press: San Diego, CA, USA, 1993.
19. Pardos, Z.; Bergner, Y.; Seaton, D.; Pritchard, D. Adapting bayesian knowledge tracing to a massive open online course in edx. *Educ. Data Min.* **2013**, *13*, 137–144.
20. Cen, H.; Koedinger, K.; Junker, B. Learning factors analysis–a general method for cognitive model evaluation and improvement. In Proceedings of the International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan, 26–30 June 2006; pp. 164–175.
21. Pavlik, P.; Cen, H.; Koedinger, K. Performance factors analysis—A new alternative to knowledge tracing. In Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK, 6–10 July 2009; pp. 531–538.
22. Desmarais, M.; Baker, R. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Model. User Adapt. Interact.* **2012**, *22*, 9–38. [CrossRef]
23. Huang, Z.; Liu, Q.; Zhai, C.; Yin, Y.; Hu, G. Exploring multi-objective exercise recommendations in online education systems. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019.
24. Liu, Q.; Tong, S.; Liu, C.; Zhao, H.; Chen, E.; Ma, H.; Wang, S. Exploiting cognitive structure for adaptive learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 627–635.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inform. Process. Syst.* **2017**, *30*, 5998–6008.
26. Khajah, M.; Lindsey, R.; Mozer, M. How deep is knowledge tracing? *arXiv* **2016**, arXiv:1604.02416.
27. Yeung, C.; Yeung, D. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *arXiv* **2018**, arXiv:1806.02180.
28. Abdelrahman, G.; Wang, Q. Knowledge tracing with sequential key-value memory networks. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019.
29. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North. Association for Computational Linguistics, Minneapolis, MN, USA, 3–5 June 2019.
30. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
31. Wang, X.; Mei, X.; Huang, Q.; Han, Z.; Huang, C. Fine-grained learning performance prediction via adaptive sparse self-attention networks. *Inf. Sci.* **2021**, *545*, 223–240. [CrossRef]
32. Zhu, J.; Yu, W.; Zheng, Z.; Huang, C.; Fung, G. Learning from Interpretable Analysis: Attention Based Knowledge Tracing. In Proceedings of the International Conference on Artificial Intelligence in Education, Ifrane, Morocco, 9–11 November 2020.
33. Choi, Y.; Lee, Y.; Cho, J.; Baek, J.; Kim, B.; Cha, Y.; Shin, D.; Bae, C.; Heo, J. Towards an appropriate query, key, and value computation for knowledge tracing. In Proceedings of the Seventh ACM Conference on Learning, Online, 12–14 August 2020; pp. 341–344.

34. Shin, D.; Shim, Y.; Yu, H.; Lee, S.; Kim, B.; Choi, Y. Saint+: Integrating temporal features for ednet correctness prediction. In Proceedings of the 11th International Learning Analytics and Knowledge Conference, Irvine, CA, USA, 11 April 2021.
35. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, 3111–3119.
36. Oord, V.; Vinyals, A.; Kavukcuoglu, K. Neural Discrete Representation Learning. In Proceedings of the International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6306–6315
37. Stamper, J.; Niculescu-Mizil, A.; Ritter, S.; Gordon, G.; Koedinger, K. Algebra 2005–2006 and Bridge to Algebra 2006–2007. In Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining Workshop, Washington, DC, USA, 25–28 July 2010.
38. Feng, M.; Hefernan, N.; Koedinger, K. Addressing the assessment challenge with an online system that tutors as it assesses. *User Model. User Adapt. Interact.* **2009**, *19*, 243–266. [CrossRef]
39. Patikorn, T.; Hefernan, N.; Baker, R. Assistments longitudinal data mining competition 2017: A preface. In Proceedings of the International Conference on Educational Data Mining, Buffalo, NY, USA, 12–15 July 2018.
40. Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
41. Gervet, T.; Koedinger, K.; Schneider, J.; Mitchell, T. When is deep learning the best approach to knowledge tracing? *J. Educ. Data Min.* **2020**, *12*, 31–54.
42. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic diferentiation in PyTorch. In Proceedings of the International Conference on Neural Information Processing Systems Workshop, Long Beach, CA, USA, 4–9 December 2017.
43. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.