Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Special Track on AI and Social Good

# Denoised Attention and Question-Augmented Representations for Knowledge Tracing

**Jiwei Deng**[1], **Youheng Bai**[1], **Mingliang Hou**[1,2*], **Teng Guo**[1], **Zitao Liu**[1] and **Weiqi Luo**[1]
[1]Guangdong Institute of Smart Education, Jinan University, Guangzhou, China
[2]TAL Education Group, Beijing, China
dengjiwei@stu.jnu.edu.cn, yhbai@stu2024.jnu.edu.cn, houmingliang@tal.com, {tengguo, liuzitao, lwq}@jnu.edu.cn

## Abstract

Knowledge tracing (KT) is an essential task in online education systems. It aims to predict the future performance of students based on their historical learning interaction data. Despite significant advancements in attention-based KT models, they still face some limitations: inaccurate input representation and excessive student forgetting modeling. These limitations often lead to the attention noise problem: the model assigns non-negligible attention weight to some information that is cognitively irrelevant in nature, thereby generating interference signals. To address this problem, we propose a novel KT model, i.e., DenoiseKT. DenoiseKT effectively models the difficulty of the questions and utilizes graph neural network to capture the complex relationship between questions, thereby refining the representations of input features. Additionally, the denoised attention mechanism introduces a weight factor to reduce the model's attention weight distribution on irrelevant information. We extensively compare DenoiseKT with 22 state-of-the-art KT models on 4 widely-used public datasets. Experimental results show that DenoiseKT can effectively solve the attention noise problem and outperform other models. The source code of DenoiseKT is available at https://pykt.org.

## 1 Introduction

Knowledge tracing (KT) is an important task in online education systems, which focuses on predicting students' future performance based on their past learning interactions. This is accomplished by modeling students' knowledge states as they engage with learning platforms like massive open online courses and intelligent tutoring systems [Piech *et al.*, 2015; Ghosh *et al.*, 2020; Shen *et al.*, 2021; Yin *et al.*, 2023; Gao *et al.*, 2025b]. Effectively addressing the KT task can help teachers better identify students who need further attention or recommend personalized learning materials to students. This is crucial to promoting the development of personalized education [Li *et al.*, 2020; Huang *et al.*, 2023].

Currently, mainstream KT models can be broadly divided into two major categories: deep sequence KT models [Medsker *et al.*, 2001] and attention-based KT models [Vaswani *et al.*, 2017]. The former (e.g., DKT [Piech *et al.*, 2015], DKT-F [Nagatani *et al.*, 2019], KQN [Lee and Yeung, 2019]) utilizes auto-regressive architectures, such as long short-term memory (LSTM) and gated recurrent unit (GRU), to capture students' interaction information at discrete timestamps, modeling temporal dependencies through iterative updates of hidden states. The latter (e.g., SAKT [Pandey and Karypis, 2019], SAINT [Choi *et al.*, 2020], AKT [Ghosh *et al.*, 2020]) employs multi-head attention mechanisms to flexibly model long-term dependencies in students' historical interaction sequences, often resulting in improved predictive accuracy. Compared to deep sequence models, attention-based models not only capture long-range dependencies more effectively but offer the advantage of parallel training. As a result, they have gradually become the dominant KT architecture in recent years.

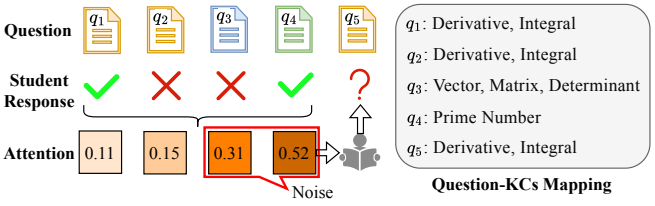



Figure 1: An illustration of the attention noise.

Although attention-based KT models have shown promising results, they face the "attention noise" problem in practical applications: the model assigns non-negligible attention weights to some information that is essentially cognitively irrelevant, thereby generating interference signals. As shown in Figure 1, for the adjacent question, although their knowledge components (KCs)[1] have no intersection, the attention mechanism may mistakenly overestimate the correlation between them and assign them non-negligible attention weights, resulting in some questions that do have cognitive connections

*Corresponding author.

[1]A knowledge component (KC) is a generalization of everyday terms like concept, principle, fact, or skill.

being ignored.

To effectively model students' forgetting behavior during cognitive processes, current mainstream attention-based KT models [Ghosh *et al.*, 2020; Im *et al.*, 2023; Yin *et al.*, 2023; Li *et al.*, 2024c] often introduce a decay mechanism that penalizes attention weights based on the temporal distance between student interaction logs—the greater the distance, the stronger the penalty—to simulate the effect of forgetting. However, the decay mechanism, to some extent, introduces noise into the attention. The decay mechanism included in the model emphasizes immediate learning behaviors, which may unintentionally lead to the neglect of learning materials with long-term cognitive significance [Salthouse *et al.*, 2006]. This is similar to the theory of "irrelevant information inhibition" in cognitive psychology, which holds that in the cognitive process, individuals need to actively suppress irrelevant information and focus on task-related signals [McNamara and McDaniel, 2004; Li *et al.*, 2023]. In KT task, if a model places too much emphasis on short-term learning behaviors while overlooking long-term cognitive development, it may struggle to accurately reflect students' evolving knowledge states. This imbalance can make it difficult to recognize students' gradual cognitive progress, thereby reducing the model's predictive accuracy. In addition, inaccurate input representation may also lead to attention noise problem. In educational scenarios, student's learning behaviors and the relationships between them are very complex. If the input of the model cannot accurately reflect these learning behaviors and the relationships between them, the attention mechanism cannot correctly focus on key information, resulting in attention noise.

To address the above attention noise problem, we present a novel KT model, called **DenoiseKT**, which focuses on the following two aspects. On the one hand, we enhance the representation by using a graph neural network to capture the complex relationships between questions, thereby obtaining a refined initial embedding representations. In addition, we also effectively model the difficulty of the questions. By combining the refined initial embedding representation and the modeling of the difficulty of the questions, we can finely encode the interaction sequence to reduce the interference of irrelevant information on the model. On the other hand, we refine the attention mechanism by introducing a weighting factor that suppresses the allocation of attention to irrelevant information. Through these two improvements, our model can solve the problem of attention noise, thereby improving the prediction performance of the model.

The main contributions of this paper are summarized as follows:

- We present the first solution to the attention noise problem in attention-based KT models, emphasizing the impact of irrelevant information on the distribution of attention weights.
- We propose the question-enhanced Rasch module (QERM) that can effectively capture the complex relationship between questions and effectively model the difficulty of questions, thereby making the feature representation more refined.
- We propose a denoised attention mechanism that can effectively reduce the model's attention weight allocation to irrelevant information.
- We conduct comprehensive quantitative and qualitative experiments to validate DenoiseKT on four public datasets, demonstrating significant performance improvements over existing 22 KT models and its effectiveness in alleviating the attention noise problem.

## 2 Related Work

### 2.1 Knowledge Tracing

KT aims to utilize historical learning interaction data from students to predict their responses to future questions. Recently, with the advancement of deep learning, this technology has been widely applied to KT tasks. We classify existing methods into deep sequence models and attention-based models based on the characteristics of KT models.

Deep sequence models employ auto-regressive architectures to capture the chronological order of student interactions. DKT [Piech *et al.*, 2015] introduced LSTM layers to assess students' knowledge mastery over time, and KQN [Lee and Yeung, 2019] built upon DKT by incorporating a skill encoder, which combines both student learning behaviors and KCs representations to further refine the predictions made by DKT. Models like DKVMN [Zhang *et al.*, 2017] leveraged memory modules for dynamic KT, and ATKT [Guo *et al.*, 2021] incorporated adversarial techniques to create perturbations that enhance the generalization ability of the model. Other extensions, such as DIMKT [Shen *et al.*, 2022] utilized multi-dimensional features to enhance predictions. However, deep sequence models often struggle with vanishing gradients and have a limited ability to capture long-range dependencies.

Attention-based models overcome many limitations of deep sequence models by utilizing self-attention mechanisms to capture long-term dependencies. SAKT [Pandey and Karypis, 2019] employed a self-attention network to model the relevance between KCs and students' past interactions, while SAINT [Choi *et al.*, 2020] introduced an encoder-decoder architecture to represent sequences of exercise and response embeddings. AKT [Ghosh *et al.*, 2020] implemented three self-attention modules, using a monotonic attention mechanism to explicitly model the phenomenon of student forgetting over time, and hybrid approaches such as SKVMN [Abdelrahman and Wang, 2019] and DTransformer [Yin *et al.*, 2023] combined attention with additional features to enhance accuracy. These attention-based models excel in parallelized training and long-range dependency awareness, positioning them as the leading choice in modern KT applications.

Despite significant advancements in attention-based KT models, they still face some limitations: inaccurate input representation and excessive student forgetting modeling. These limitations often lead to the attention noise problem, where the model assigns non-negligible attention weight to some information that is cognitively irrelevant in nature, thereby generating interference signals.

### 2.2 Attention Noise

Attention noise refers to the phenomenon that the model based on the attention mechanism assigns non-negligible attention weights to segments of the input sequence that are either irrelevant or less important to the task at hand [Gao *et al.*, 2025a]. As a result, the model fails to concentrate on the most crucial information in the sequence, instead incorporating irrelevant information into the decision-making process, which reduces the model's ability to capture the key patterns needed for accurate predictions or task execution, ultimately leading to a decline in model performance.

Attention noise has been widely recognized as a key problem affecting model performance. In recent years, many studies have tried to solve this problem from different angles. For example, [Zhao *et al.*, 2019] indirectly alleviates attention noise by designing sparse attention patterns to focus on relevant parts. [Ye *et al.*, 2024] proposed a differential attention mechanism to eliminate attention noise by calculating the attention score as the difference between two independent softmax attention maps.

However, these methods are not applicable to KT tasks, as attention noise is influenced by the modeling of student forgetting, often resulting in the noise being assigned more weight than the relevant information.

## 3 The Framework of DenoiseKT

### 3.1 Problem Statement

Given an arbitrary question $q_*$, KT's objective is to predict the probability of a student answering $q_*$ correctly based on the student's previous interaction data. More specifically, the previous interaction sequence of each student is represented as $S = \{< q_1, \mathcal{C}_1, r_1 >, < q_2, \mathcal{C}_2, r_2 >, ..., < q_t, \mathcal{C}_t, r_t >\}$, where $q_t$ denotes the question answered by the student at time $t$, $\mathcal{C}_t$ represents the associated KC set to $q_t$ and $r_t$ is the binary response indicating whether the student's response to the question is correct ($r_t$ = 1) or incorrect ($r_t$ = 0) . We would like to estimate the probability of the student answering the question $q_{t+1}$ correctly.

### 3.2 Framework Overview

DenoiseKT's overall framework is depicted in Figure 2. In this section, we provide a detailed introduction to DenoiseKT, which consists of three components: (1) interaction representation module that uses graph-enhanced question representation and incorporates question difficulty into the representation; (2) denoised attention module that effectively solves the problem of attention noise to better extract knowledge state from students' past learning history; (3) prediction module that uses a two-layer fully connected network to make prediction.

### 3.3 Interaction Representation Module

Effectively representing questions in interaction sequences is crucial to the success of the KT model. However, there are often complex and multi-level relationships between questions, making it difficult to comprehensively capture and model them using traditional methods alone. These relationships are not merely superficial adjacent connections (some questions may involve the same KC, thereby forming direct associations). Furthermore, there may exist deeper high-order relationships. For example, while question 1 and question 2 share a common KC, and question 2 and question 3 also share a KC, even though question 1 and question 3 lack direct KC connections, there may still exist inherent connections between them through this indirect chain of associations. Traditional models typically only handle the direct associations of these questions and struggle to capture more complex high-order relationships, which have a significant impact on the representation of questions.

To address the above problem, we utilize a graph convolutional network (GCN) to capture high-order relationships among questions to enhance question representations [Liu *et al.*, 2024; Sun *et al.*, 2023; Li *et al.*, 2024a; Li *et al.*, 2024b].

Given a set of questions $\mathcal{Q}$ and KCs $\mathcal{C}$, we construct a question-KC bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{U})$, where $\mathcal{V} = (\mathcal{Q} \cup \mathcal{C})$ and $\mathcal{U} = \{(q, c) | q \in \mathcal{Q}, c \in \mathcal{C}\}$ denote the sets of nodes and edges, respectively. We then use GCN combined with the question-KC bipartite graph to enhance question representations. The latent representation $\mathbf{e}_{q_t}$ of question $q_t$ is computed as follows:

$$\mathbf{W}_q^{(1)} = \mathbf{A}\mathbf{W}_q^{(0)}\mathbf{W} + \mathbf{b}$$

$$\mathbf{e}_{q_t} = \mathbf{W}_q^{(1)} \cdot \mathbf{o}_{q_t}$$

where $\mathbf{A}$ is the adjacency matrix of $\mathcal{G}$. $\mathbf{W}_q \in \mathbb{R}^{d \times n_q}$ is the embedding matrix of question. $\mathbf{o}_*$ is the original one-hot vector of $*$. $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are learnable linear transformation operations.

Additionally, even for questions that cover the same KC, there may be significant differences in difficulty. For instance, two questions that both assess fraction addition: one is a simple addition with the same denominator, while the other involves more complex operations with different denominators, students' performance on these two questions will demonstrate a significant difference. If we judge students' performance solely based on whether they have mastered a certain type of question, while ignoring the difficulty of the question, the model may make incorrect inferences. Inspired by the classic Rasch model [Rasch, 1993], which models individual ability and item difficulty separately, we incorporate question difficulty into the question representation. After obtaining the latent representation $\mathbf{e}_{q_t}$ of question $q_t$, we enhance it by introducing a difficulty parameter to better capture the underlying properties of the question. More specifically, the $t$-th representations of question and interaction are represented as follows:

$$\mathbf{d}_{q_t} = \mathbf{W}_d \cdot \mathbf{o}_{q_t}; \mathbf{e}_{c_t} = \frac{1}{|\mathcal{C}_t|} \sum_{j=1}^{|\mathcal{C}_t|} \mathbf{W}_c \cdot \mathbf{o}_{c_{t_j}}$$

$$\mathbf{e}_{r_t} = \mathbf{W}_r \cdot \mathbf{o}_{r_t}$$

$$\mathbf{x}_t = \mathbf{e}_{q_t} \oplus \mathbf{d}_{q_t} \odot \mathbf{e}_{c_t}; \quad \mathbf{y}_t = \mathbf{e}_{q_t} \oplus \mathbf{e}_{r_t}$$

where $\mathbf{d}_{q_t}$ represents the question $q_t$ difficulty. $\mathbf{e}_{c_t}$ and $\mathbf{e}_{r_t}$ denote the latent representations of KC set $\mathcal{C}_t$ and student response $r_t$ on question $q_t$. $|\mathcal{C}_t|$ is the total number of KCs associated with question $q_t$. $\mathbf{W}_d \in \mathbb{R}^{d \times n_q}$, $\mathbf{W}_c \in \mathbb{R}^{d \times n_c}$ and

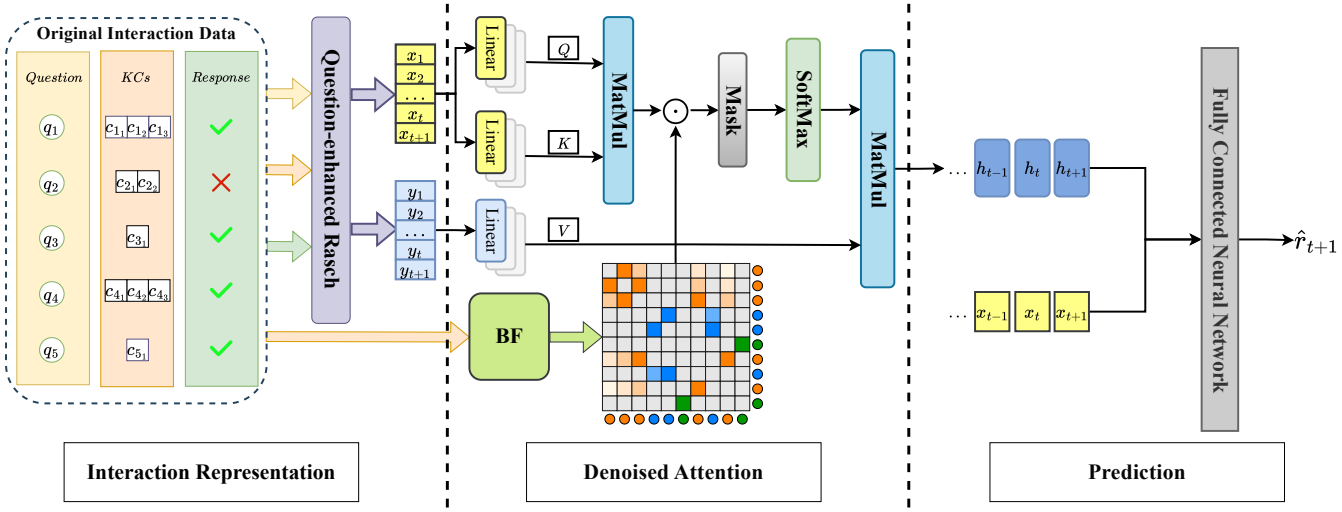



Figure 2: The overview of the proposed DenoiseKT framework.

$\mathbf{W}_r \in \mathbb{R}^{d\times 2}$ represent the difficulty embedding matrix, the KC embedding matrix, and the response embedding matrix. $\oplus$ and $\odot$ represent the addition operators and element-wise product respectively.

The above is a detailed overview of the QERM we proposed. This module can not only characterize the complex relationships between questions, but also reasonably model the difficulty differences of questions so as to represent questions more accurately.

### 3.4 Denoised Attention Mechanism

In the research field of KT and educational measurement, students' cognitive behaviors are closely linked to the questions they answer. When answering questions, students tend to focus on those questions that are most similar in content and closest in time to the current question, and answer based on this information. This process conforms to the laws of human cognition, because the human brain typically processes information through associations, especially when confronted with new problems, it automatically recalls similar situations and experiences.

In real cognitive processes, the factors that have the most significant impact on a student's current cognitive behavior are often those historical questions that are most relevant and closest to the current question. However, existing attention-based models KT models tend to assign non-negligible attention scores to questions that are not closely related to the current question, due to inaccurate input representations and the decay mechanisms introduced to account for student forgetting. These results in unnecessary interference and prevents the model from focusing on truly relevant information. These non-negligible irrelevant attention scores are referred to as attention noise. This noise affects the model's ability to accurately track and analyze the student's learning process, thereby reducing the model's overall effectiveness. To solve the attention noise problem, we design a denoised attention mechanism that multiplies the attention scores by weight factor. This enables the model to effectively focus on truly relevant information. Specifically, the retrieved knowledge state $h_{t+1}$ at the $(t+1)$-th timestamp is determined using the following formula:

$$\mathbf{Q} = L(\mathbf{x}_{t+1}); \mathbf{K} = L(\mathbf{x}_1, \ldots, \mathbf{x}_t); \mathbf{V} = L(\mathbf{y}_1, \ldots, \mathbf{y}_t)$$

$$\mathbf{h}_{t+1} = softmax(\mathbf{BF} \cdot \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}) \cdot \mathbf{V}$$

where $L(\cdot)$, $\mathbf{K}^T$ and $d$ denote the linear operation, the transpose of $\mathbf{K}$ and the dimension of $\mathbf{K}$ respectively. $\mathbf{BF}$ is the weight factor that is calculated based on the similarity and distance between questions in the interaction history. Specifically, each element of $\mathbf{BF}$ is computed as follows:

$$sim_{ij} = |i-j| \cdot \mathbf{1}_{\{q_i \simeq q_j\}}$$

$$bf_{ij} = \begin{cases} 1+\beta^{sim_{ij}}, & sim_{ij} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

where $bf_{ij}$ denotes the element at the $(i)$-th row and $(j)$-th column of $\mathbf{BF}$. $\mathbf{1}_{\{q_i \simeq q_j\}}$ means it is 1 when $q_i$ and $q_j$ share the same KC, and 0 otherwise. $\beta$ is a hyperparameter greater than 0 and less than 1.

### 3.5 Prediction Module

To predict the outcomes of students over questions, we employ a two-layer fully connected neural network following the attention layers. The attention layers capture a comprehensive representation of the student's behavioral sequence. After passing through the feed-forward network in the attention module, this representation, denoted as $\mathbf{h}_{t+1}$, functions as a summary of the knowledge state. This is subsequently merged with the contextualized embedding of the current question $\mathbf{x}_{t+1}$, creating the input for the prediction layer. The prediction layer calculates the probability of a correct response $\hat{\mathbf{r}}_{t+1}$ using the following formulation:

$$\hat{\mathbf{r}}_{t+1} = \sigma(\eta(\mathbf{W}_2 \cdot \eta(\mathbf{W}_1 \cdot [\mathbf{h}_{t+1}; \mathbf{x}_{t+1}] + \mathbf{b}_1) + \mathbf{b}_2))$$

where $\sigma(\cdot)$, $\eta(\cdot)$ denote Sigmoid and ReLU functions. $\mathbf{W}_1 \in \mathbb{R}^{d\times 2d}$, $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$, $\mathbf{b}_1 \in \mathbb{R}^d$, $\mathbf{b}_2 \in \mathbb{R}^d$ are learnable parameters.

We optimize the prediction function by minimizing the binary cross-entropy loss between the ground-truth response $\mathbf{r}_{t+1}$ and the prediction probability $\hat{\mathbf{r}}_{t+1}$ as follows:

$$\mathcal{L} = -\sum_t (\mathbf{r}_{t+1} \log \hat{\mathbf{r}}_{t+1} + (1 - \mathbf{r}_{t+1}) \log(1 - \hat{\mathbf{r}}_{t+1}))$$

This predictive framework allows the model to incorporate information from both previous interactions and current contexts, ensuring precise estimates of students' knowledge states and response probabilities.

# 4 Experiment

To evaluate the performance of our proposed DenoiseKT framework, we conduct extensive experiments to address the following research questions:

- **RQ1**: Can DenoiseKT outperform existing KT models in terms of prediction performance?
- **RQ2**: How does each component of DenoiseKT contribute to its overall effectiveness?
- **RQ3**: Can DenoiseKT effectively solve the attention noise problem?

## 4.1 Datasets

In this paper, we evaluated the performance of our model on four widely used publicly available datasets.

- **ASSISTments2009 (ASSIST2009)**[2]: This dataset is collected from the ASSISTment online tutoring platform during the 2009-2010 school year and focuses on math exercises. It has been a standard benchmark for KT methods over the past decade. The dataset includes 346,860 interactions, 4,217 students, 17,737 questions, and 123 KCs, with the maximum number of KCs in a single question being 4.
- **NeurIPS2020 Education Challenge (NIPS34)**[3]: This dataset is provided by NeurIPS 2020 Education Challenge. It contains students' responses to mathematics questions from Eedi. The dataset includes data from Task 3 and Task 4, with 1,399,470 interactions, 4,918 students, 948 questions, 57 KCs, and the maximum number of KCs in a single question is 2.
- **XES3G5M**[4]: This large-scale dataset derived from a Chinese online mathematics learning platform, documenting the learning performance of third-grade students in mathematics. The dataset contains 5,549,635 interactions involving 18,066 students across 7,652 math questions. It includes rich auxiliary information,such as 865 KCs with hierarchical relationships, question types, textual content and analysis, as well as timestamps of student responses. In terms of the number of KCs and the richness of contextual information, this dataset is currently the largest in the mathematics domain.
- **EdNet**[5]: This dataset is collected over two years by Santa, an AI tutoring service with over 780,000 users in Korea. Given the large volume, we randomly select a subset of student records for model evaluation. The dataset includes 597,042 interactions, 4,999 students, 11,901 questions, and 188 KCs, with each question associated with up to 7 KCs.

To ensure reproducibility in our experiments, we rigorously follow the data pre-processing steps suggested in [Liu *et al.*, 2022b]. We filter out student sequences that are shorter than 3 interactions. Data statistics are summarized in Table 1.

| Dataset | # of interactions | # of students | # questions | # of KCs |
|---|---|---|---|---|
| ASSIST2009 | 346,860 | 4,217 | 17,737 | 123 |
| NIPS34 | 1,399,470 | 4,918 | 948 | 57 |
| XES3G5M | 5,549,635 | 18,066 | 7,652 | 865 |
| EdNet | 597,042 | 4,999 | 11,901 | 188 |

Table 1: Data statistics of 4 widely used datasets.

## 4.2 Baselines

To assess the performance of DenoiseKT, we compared it with 22 state-of-the-art KT models as follows: DKT [Piech *et al.*, 2015], DKT+ [Yeung and Yeung, 2018], DKT-F [Nagatani *et al.*, 2019], KQN [Lee and Yeung, 2019], DKVMN [Zhang *et al.*, 2017], ATKT [Guo *et al.*, 2021], GKT [Nakagawa *et al.*, 2019], SAKT [Pandey and Karypis, 2019], SAINT [Choi *et al.*, 2020], AKT [Ghosh *et al.*, 2020], SKVMN [Abdelrahman and Wang, 2019], HawkesKT [Wang *et al.*, 2021], Deep-IRT [Yeung, 2019], DIMKT [Shen *et al.*, 2022], qDKT [Sonkar *et al.*, 2020], AT-DKT [Liu *et al.*, 2023], simpleKT [Liu *et al.*, 2022a], QIKT [Chen *et al.*, 2023], RKT [Pandey and Srivastava, 2020], FoLiBiKT [Im *et al.*, 2023], DTransformer [Yin *et al.*, 2023], extraKT [Li *et al.*, 2024c].

It is important to note that the baseline models selected in this paper are mostly derived from articles published in top venues on AI/ML or education. Furthermore, when comparing these baseline models, the implementations were based on the authors' publicly available code and reproduced following the configuration settings in pyKT [Liu *et al.*, 2022b]. All methods underwent optimal hyperparameter tuning using Weights&Bias[6], ensuring the fairness of the comparison results. Moreover, the hyper-parameter tuning tool Weights&Biases utilizes Bayesian search to automatically identify the optimal parameter combinations. For each model, the tool explores approximately hundreds of combinations during the tuning process. This automated approach ensures that all methods achieve optimal parameters on each dataset.

[2]https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010

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

[4]https://github.com/ai4ed/XES3G5M

[5]https://github.com/riiid/ednet

[6]https://wandb.ai/site/

| Model | AUC | | | | ACC | | | |
|---|---|---|---|---|---|---|---|---|
| | ASSIST2009 | NIPS34 | XES3G5M | EdNet | ASSIST2009 | NIPS34 | XES3G5M | EdNet |
| DKT | 0.7541 ± 0.0011 | 0.7689 ± 0.0002 | 0.7852 ± 0.0006 | 0.6133 ± 0.0006 | 0.7244 ± 0.0014 | 0.7032 ± 0.0004 | 0.8173 ± 0.0002 | 0.6462 ± 0.0028 |
| DKT+ | 0.7547 ± 0.0017 | 0.7696 ± 0.0002 | 0.7861 ± 0.0002 | 0.6189 ± 0.0012 | 0.7248 ± 0.0009 | 0.7039 ± 0.0004 | 0.8178 ± 0.0001 | 0.6571 ± 0.0019 |
| DKT-F | - | 0.7733 ± 0.0003 | 0.7940 ± 0.0006 | 0.6168 ± 0.0019 | - | 0.7076 ± 0.0002 | 0.8209 ± 0.0003 | 0.6402 ± 0.0021 |
| KQN | 0.7477 ± 0.0011 | 0.7684 ± 0.0003 | 0.7793 ± 0.0006 | 0.6111 ± 0.0022 | 0.7228 ± 0.0009 | 0.7028 ± 0.0001 | 0.8152 ± 0.0002 | 0.6422 ± 0.0043 |
| DKVMN | 0.7473 ± 0.0006 | 0.7673 ± 0.0004 | 0.7792 ± 0.0004 | 0.6158 ± 0.0022 | 0.7199 ± 0.0010 | 0.7016 ± 0.0005 | 0.8155 ± 0.0001 | 0.6444 ± 0.0030 |
| ATKT | 0.7470 ± 0.0008 | 0.7665 ± 0.0001 | 0.7783 ± 0.0004 | 0.6065 ± 0.0003 | 0.7208 ± 0.0009 | 0.7013 ± 0.0002 | 0.8145 ± 0.0002 | 0.6369 ± 0.0009 |
| GKT | 0.7424 ± 0.0021 | 0.7689 ± 0.0024 | 0.7727 ± 0.0006 | 0.6223 ± 0.0017 | 0.7153 ± 0.0032 | 0.7014 ± 0.0028 | 0.8135 ± 0.0004 | 0.6625 ± 0.0064 |
| SAKT | 0.7246 ± 0.0017 | 0.7517 ± 0.0005 | 0.7693 ± 0.0008 | 0.6072 ± 0.0018 | 0.7063 ± 0.0018 | 0.6879 ± 0.0004 | 0.8124 ± 0.0002 | 0.6391 ± 0.0041 |
| SAINT | 0.6958 ± 0.0023 | 0.7873 ± 0.0007 | 0.8074 ± 0.0007 | 0.6614 ± 0.0019 | 0.6936 ± 0.0034 | 0.7180 ± 0.0006 | 0.8177 ± 0.0006 | 0.6522 ± 0.0024 |
| AKT | 0.7853 ± 0.0017 | 0.8033 ± 0.0003 | 0.8207 ± 0.0008 | 0.6721 ± 0.0022 | 0.7392 ± 0.0021 | 0.7323 ± 0.0005 | 0.8273 ± 0.0007 | 0.6655 ± 0.0042 |
| SKVMN | 0.7332 ± 0.0009 | 0.7513 ± 0.0005 | 0.7514 ± 0.0005 | 0.6182 ± 0.0114 | 0.7156 ± 0.0012 | 0.6885 ± 0.0005 | 0.8075 ± 0.0003 | 0.6555 ± 0.0152 |
| HawkesKT | 0.7224 ± 0.0006 | 0.7767 ± 0.0010 | 0.7921 ± 0.0007 | 0.6837 ± 0.0016 | 0.7046 ± 0.0008 | 0.7110 ± 0.0007 | 0.8188 ± 0.0003 | 0.6917 ± 0.0013 |
| Deep-IRT | 0.7465 ± 0.0006 | 0.7672 ± 0.0006 | 0.7785 ± 0.0005 | 0.6173 ± 0.0008 | 0.7195 ± 0.0004 | 0.7014 ± 0.0008 | 0.8150 ± 0.0002 | 0.6457 ± 0.0033 |
| DIMKT | 0.7717 ± 0.0011 | 0.8030 ± 0.0002 | 0.8220 ± 0.0002 | 0.6748 ± 0.0030 | 0.7354 ± 0.0019 | 0.7312 ± 0.0005 | 0.8291 ± 0.0006 | 0.6699 ± 0.0038 |
| qDKT | 0.7016 ± 0.0049 | 0.7995 ± 0.0008 | 0.8225 ± 0.0002 | 0.6987 ± 0.0010 | 0.6787 ± 0.0039 | 0.7299 ± 0.0007 | 0.8301 ± 0.0000 | 0.6922 ± 0.0004 |
| AT-DKT | 0.7555 ± 0.0005 | 0.7816 ± 0.0002 | 0.7932 ± 0.0004 | 0.6249 ± 0.0020 | 0.7250 ± 0.0007 | 0.7146 ± 0.0002 | 0.8198 ± 0.0004 | 0.6512 ± 0.0039 |
| simpleKT | 0.7744 ± 0.0018 | 0.8035 ± 0.0000 | 0.8163 ± 0.0006 | 0.6599 ± 0.0027 | 0.7320 ± 0.0012 | 0.7328 ± 0.0001 | 0.8246 ± 0.0005 | 0.6557 ± 0.0029 |
| QIKT | 0.7878 ± 0.0024 | 0.8044 ± 0.0005 | 0.8222 ± 0.0006 | 0.7271 ± 0.0012 | 0.7381 ± 0.0014 | 0.7333 ± 0.0005 | 0.8300 ± 0.0005 | 0.7082 ± 0.0016 |
| RKT | 0.7628 ± 0.0070 | 0.7966 ± 0.0011 | 0.8224 ± 0.0007 | 0.7226 ± 0.0005 | 0.7289 ± 0.0062 | 0.7264 ± 0.0010 | 0.8294 ± 0.0004 | 0.7045 ± 0.0022 |
| FoLiBiKT | 0.7838 ± 0.0013 | 0.8033 ± 0.0002 | 0.8214 ± 0.0007 | 0.6747 ± 0.0036 | 0.7393 ± 0.0010 | 0.7323 ± 0.0001 | 0.8271 ± 0.0006 | 0.6676 ± 0.0028 |
| DTransformer | 0.7725 ± 0.0025 | 0.7944 ± 0.0003 | 0.8144 ± 0.0006 | 0.6736 ± 0.0028 | 0.7295 ± 0.0017 | 0.7295 ± 0.0007 | 0.8248 ± 0.0004 | 0.6665 ± 0.0017 |
| extraKT | 0.7824 ± 0.0013 | 0.8045 ± 0.0003 | 0.8200 ± 0.0008 | 0.6730 ± 0.0025 | 0.7355 ± 0.0017 | **0.7340 ± 0.0004** | 0.8263 ± 0.0010 | 0.6668 ± 0.0016 |
| DenoiseKT | **0.7898 ± 0.0013** | **0.8046 ± 0.0002** | **0.8282 ± 0.0004** | **0.7353 ± 0.0012** | **0.7427 ± 0.0007** | 0.7330 ± 0.0007 | **0.8319 ± 0.0003** | **0.7131 ± 0.0020** |

Table 2: Performance comparisons in terms of AUC and ACC on four datasets. The best performance is highlighted in bold, and the second-best performance is underlined.

## 4.3 Experimental Setting

In order to fairly evaluate the performance of KT models, all models are trained and evaluated on student interaction sequences of fixed length 200. For each dataset, we use 20% of all student sequences for the test set, and perform standard 5-fold cross validation on the rest 80% of all student sequences. We use the Adam optimizer to train our model. The maximum of the training epochs is set to 200, and we choose to use early stopping when the performance is not improved after 10 epochs. The embedding dimension, the hidden state dimension, the two dimension of the feed-forward neural network layers are set to [64, 128], the learning rate, random seed, dropout rate and hyper-parameter $\beta$ are set to [1e-3, 2e-3, 1e-4], [42, 3407], [0.1, 0.2, 0.3, 0.4, 0.5] and [0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99] respectively, the number of blocks and attention heads are set to [1, 2, 4] and [4, 8]. Our model is implemented in PyTorch [Imambi *et al.*, 2021] and trained on a cluster of Linux servers with the NVIDIA RTX 3090 GPU device. Aligned with previous work [Liu *et al.*, 2022b; Ghosh *et al.*, 2020; Liu *et al.*, 2022a], we use area under the curve (AUC) as the main evaluation metric and accuracy (ACC) as the secondary evaluation metric.

## 4.4 Results

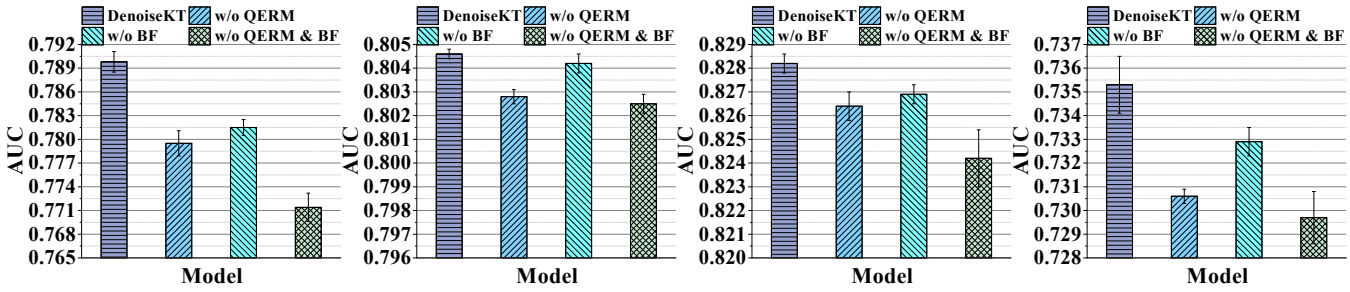
### Overall Performance (RQ1)

Table 2 shows the overall performance of all models in the four datasets. We calculate the mean and standard deviation of the AUC and ACC across 5 folds. According to Table 2, we have the following observations: (1) on the main evaluation metrics, our proposed model DenoiseKT outperforms all state-of-the-art models. Compared to the second-best performing model, the DenoiseKT model performs 0.20% better on ASSIST2009, 0.01% better on NIPS34, 0.57% better on XES3G5M, and 0.82% better on EdNet. On the secondary evaluation metrics, our proposed model DenoiseKT outperforms almost all baselines (except QIKT and extraKT on the NIPS34 dataset) , with scores of 0.7427 (ASSIST2009), 0.7330 (NIPS34), 0.8319 (XES3G5M), and 0.7131 (EdNet). These results indicate the effectiveness of DenoiseKT; (2) compared to all attention-based KT models, i.e., SAKT, SAINT, AKT and simpleKT, our model has the best performance on all four datasets. This indicates our denoised attention mechanism and QERM allow attention-based KT models to effectively deal with the attention noise problem that improves the predictive performance.

Although the AUC improvements shown in Table 2 may appear modest, with increases of less than 1% on some datasets, these gains are significant, especially when evaluated with a context window size of 200. Recent benchmarking studies have highlighted that many reported advancements in KT models are unreliable, often stemming from flawed evaluation methods, with overall prediction performance having improved by only 3.5% since 2015. In contrast, our study strictly adhered to the rigorous evaluation protocol proposed by pyKT [Liu *et al.*, 2022b] and conducted an exhaustive hyper-parameter search for all baseline models to ensure reliable and meaningful comparisons.

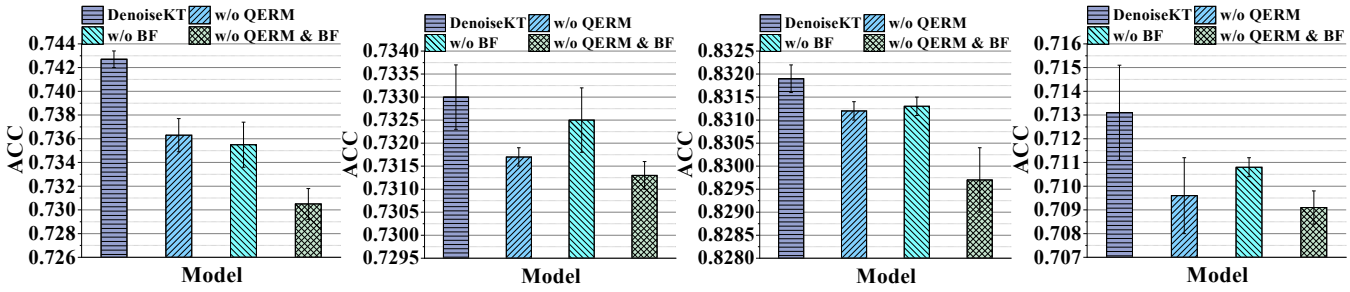
### Ablation Study (RQ2)

In order to verify the effectiveness of each design component in the DenoiseKT model, we constructed three variants of the DenoiseKT model for ablation experiments, as shown in Figures 3 and 4. Specifically, for variant DenoiseKT w/o QERM, we removed the QERM in the question representation part, that is to say we no longer use GCN to capture the complex relationship between questions to enhance question representation and also cancel the effective modeling of problem difficulty. We use the standard method commonly used in KT research to represent the question. For variant DenoiseKT w/o BF, we removed the weight factor BF used to solve the noise problem in the denoised attention mechanism. For variant DenoiseKT w/o QERM & BF, we removed both the QERM and the weight factor BF.




(a) ASSIST2009. (b) NIPS34. (c) XES3G5M. (d) EdNet.

Figure 3: Comparison of AUC values for DenoiseKT and its variants across datasets.




(a) ASSIST2009. (b) NIPS34. (c) XES3G5M. (d) EdNet.

Figure 4: Comparison of ACC values for DenoiseKT and its variants across datasets.

It can be observed that compared with the complete DenoiseKT model, all other ablation variants perform worse on the four datasets. Specifically, (1) the variant without QERM (DenoiseKT w/o QERM) had different degrees of performance degradation on all datasets, among which the performance of ASSIST2009 was the most obvious, with AUC down by 1.3%. This emphasizes the importance of capturing the complex relationship between problems and effectively modeling the difficulty of problems; (2) the variant without BF (DenoiseKT w/o BF) also leads to a decrease in performance, which shows that reducing the generation of attention noise is crucial for accurate prediction; (3) compared with other variants, the variant without QERM and BF (DenoiseKT w/o QERM & BF) shows the worst results.

Overall, the absence of any component will lead to a decrease in model performance, which shows that each module in the DenoiseKT model plays an important role in the KT task.

**Visualization (RQ3)**

To intuitively demonstrate the difference between regular dot-product attention and denoised attention, we visualize the attention scores of both regular dot-product attention and denoised attention, as shown in Figure 5. From Figure 5, we have the following observations: our denoised attention can capture the information of similar problems well, while regular dot product attention cannot capture the information of similar problems well. For example, when predicting the 9th question, denoised attention focuses on information from questions 4, 3, 2, and 1, while regular dot-product attention emphasizes information from questions 8, 7, 6 and 5. This demonstrates that our model can efficiently tackle the issue of attention noise.

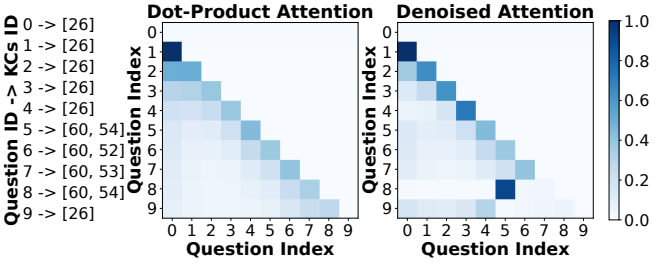



Figure 5: Visualization of both dot-product attention and denoised attention. The question index represents questions answered by a specific student, and index 0 represents the first question.

## 5 Conclusion

This paper presents DenoiseKT, a KT model designed to address attention noise by introducing two key components: a QERM for refined question representation, and a denoised attention mechanism to suppress irrelevant signals. Experiments on four public datasets confirm that both components contribute to improved prediction accuracy. Beyond performance gains, DenoiseKT offers a generalizable and cognitively grounded framework for calibrating attention in educational modeling, paving the way for more interpretable and reliable KT systems.

## Acknowledgements

This work was supported in part by National Key R&D Program of China, under Grant No. 2023YFC3341200; in part by Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University (2022LSYS003) and in part by Beijing Municipal Science and Technology Project under Grant No. Z241100001324011.

## References

[Abdelrahman and Wang, 2019] Ghodai Abdelrahman and Qing Wang. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2019.

[Chen *et al.*, 2023] Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14196–14204, 2023.

[Choi *et al.*, 2020] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the 7th ACM Conference on Learning at Scale*, pages 341–344, 2020.

[Gao *et al.*, 2025a] Weibo Gao, Qi Liu, Rui Li, Yuze Zhao, Hao Wang, Linan Yue, Fangzhou Yao, and Zheng Zhang. Denoising programming knowledge tracing with a code graph-based tuning adaptor. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 354–365, 2025.

[Gao *et al.*, 2025b] Weibo Gao, Qi Liu, Linan Yue, Fangzhou Yao, Zhenya Huang, Zheng Zhang, and Rui Lv. Boxcd: Leveraging contrastive probabilistic box embedding for effective and efficient learner modeling. In *Proceedings of the ACM on Web Conference 2025*, pages 3660–3671, 2025.

[Ghosh *et al.*, 2020] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2330–2339, 2020.

[Guo *et al.*, 2021] Xiaopeng Guo, Zhijie Huang, Jie Gao, Mingyu Shang, Maojing Shu, and Jun Sun. Enhancing knowledge tracing via adversarial training. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 367–375, 2021.

[Huang *et al.*, 2023] Shuyan Huang, Zitao Liu, Xiangyu Zhao, Weiqi Luo, and Jian Weng. Towards robust knowledge tracing models via k-sparse attention. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 2441–2445, 2023.

[Im *et al.*, 2023] Yoonjin Im, Eunseong Choi, Heejin Kook, and Jongwuk Lee. Forgetting-aware linear bias for attentive knowledge tracing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3958–3962, 2023.

[Imambi *et al.*, 2021] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.

[Lee and Yeung, 2019] Jinseok Lee and Dit-Yan Yeung. Knowledge query network for knproceedings of the 26th international conference on world wide webowledge tracing: How knowledge interacts with skills. In *Proceedings of the 9th International Conference on Learning Analytics and Knowledge*, pages 491–500, 2019.

[Li *et al.*, 2020] Hang Li, Wenbiao Ding, and Zitao Liu. Identifying at-risk k-12 students in multimodal online environments: A machine learning approach. *International Educational Data Mining Society*, 2020.

[Li *et al.*, 2023] Qing Li, Xin Yuan, Sannyuya Liu, Lu Gao, Tianyu Wei, Xiaoxuan Shen, and Jianwen Sun. A genetic causal explainer for deep knowledge tracing. *IEEE Transactions on Evolutionary Computation*, 2023.

[Li *et al.*, 2024a] Ming Li, Zhao Li, Changqin Huang, Yunliang Jiang, and Xindong Wu. Edugraph: Learning path-based hypergraph neural networks for mooc course recommendation. *IEEE Transactions on Big Data*, 2024.

[Li *et al.*, 2024b] Ming Li, Siwei Zhou, Yuting Chen, Changqin Huang, and Yunliang Jiang. Educross: Dual adversarial bipartite hypergraph learning for cross-modal retrieval in multimodal educational slides. *Information Fusion*, 109:102428, 2024.

[Li *et al.*, 2024c] Xueyi Li, Youheng Bai, Ying Zheng, Mingliang Hou, Bojun Zhan, Yaying Huang, Zitao Liu, Boyu Gao, and Weiqi Luo. Extending context window of attention based knowledge tracing models via length extrapolation. In *Proceedings of the 27th European Conference on Artificial Intelligence*, 2024.

[Liu *et al.*, 2022a] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *Proceedings of the 11th International Conference on Learning Representations*, 2022.

[Liu *et al.*, 2022b] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. pyKT: a python library to benchmark deep learning based knowledge tracing models. *Advances in Neural Information Processing Systems*, 35:18542–18555, 2022.

[Liu *et al.*, 2023] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Boyu Gao, Weiqi Luo, and Jian Weng. Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the 32th International Conference on World Wide Web*, pages 4178–4187, 2023.

[Liu *et al.*, 2024] Shengyingjie Liu, Zongkai Yang, Sannyuya Liu, Ruxia Liang, Jianwen Sun, Qing Li, and Xiaoxuan Shen. Hyperbolic embedding of discrete evolution

graphs for intelligent tutoring systems. *Expert Systems with Applications*, 241:122451, 2024.

[McNamara and McDaniel, 2004] Danielle S McNamara and Mark A McDaniel. Suppressing irrelevant information: Knowledge activation or inhibition? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):465, 2004.

[Medsker *et al.*, 2001] Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and Applications*, 5(64-67):2, 2001.

[Nagatani *et al.*, 2019] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *Proceedings of the 28th International Conference on World Wide Web*, pages 3101–3107, 2019.

[Nakagawa *et al.*, 2019] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 156–163, 2019.

[Pandey and Karypis, 2019] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. In *12th International Conference on Educational Data Mining*, pages 384–389, 2019.

[Pandey and Srivastava, 2020] Shalini Pandey and Jaideep Srivastava. RKT: relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 1205–1214, 2020.

[Piech *et al.*, 2015] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 28, 2015.

[Rasch, 1993] Georg Rasch. Probabilistic models for some intelligence and attainment tests. 1993.

[Salthouse *et al.*, 2006] Timothy A Salthouse, John R Nesselroade, and Diane E Berish. Short-term variability in cognitive performance and the calibration of longitudinal change. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 61(3):P144–P151, 2006.

[Shen *et al.*, 2021] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1452–1460, 2021.

[Shen *et al.*, 2022] Shuanghong Shen, Zhenya Huang, Qi Liu, Yu Su, Shijin Wang, and Enhong Chen. Assessing student's dynamic knowledge state by exploring the question difficulty effect. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 427–437, 2022.

[Sonkar *et al.*, 2020] Shashank Sonkar, Andrew E Waters, Andrew S Lan, Phillip J Grimaldi, and Richard G Baraniuk. qDKT: Question-centric deep knowledge tracing. In *Proceedings of the 13th International Conference on Educational Data Mining*, 2020.

[Sun *et al.*, 2023] Jianwen Sun, Shangheng Du, Zhi Liu, Fenghua Yu, Sannyuya Liu, and Xiaoxuan Shen. Weighted heterogeneous graph-based three-view contrastive learning for knowledge tracing in personalized e-learning systems. *IEEE Transactions on Consumer Electronics*, 70(1):2838–2847, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2021] Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, and Shaoping Ma. Temporal cross-effects in knowledge tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 517–525, 2021.

[Ye *et al.*, 2024] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

[Yeung and Yeung, 2018] Chun-Kit Yeung and Dit-Yan Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the 15th Annual ACM Conference on Learning at Scale*, pages 1–10, 2018.

[Yeung, 2019] Chun-Kit Yeung. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. *arXiv e-prints*, pages arXiv–1904, 2019.

[Yin *et al.*, 2023] Yu Yin, Le Dai, Zhenya Huang, Shuanghong Shen, Fei Wang, Qi Liu, Enhong Chen, and Xin Li. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the 32th International Conference on World Wide Web*, pages 855–864, 2023.

[Zhang *et al.*, 2017] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 765–774, 2017.

[Zhao *et al.*, 2019] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. Explicit sparse transformer: Concentrated attention through explicit selection, 2019.