ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240651

Extending Context Window of Attention Based Knowledge Tracing Models via Length Extrapolation

Xueyi Li[a], Youheng Bai[a], Teng Guo[a], Ying Zheng[a], Mingliang Hou[b,*], Bojun Zhan[a], Yaying Huang[c], Zitao Liu[a], Boyu Gao[a] and Weiqi Luo[a]

[a]Guangdong Institute of Smart Education, Jinan University, Guangzhou, China
[b]TAL Education Group, Beijing, China
[c]The Primary School attached to Jinan University, Jinan University, Guangzhou, China

Abstract. Knowledge tracing (KT) is a prediction task that aims to predict students' future performance based on their past learning data. The rapid progress in attention mechanisms has led to the emergence of various high-performing attention based KT models. However, in online or personalized education settings, students' varying learning paths result in different lengths of student interaction sequences, which poses a significant challenge for attention based KT models as their context window sizes are fixed during both training and prediction stages. We refer to this as *the length extrapolation of KT model*. In this paper, we propose **extraKT** to facilitate better extrapolation that learn from student interactions with a short context window and continue to perform well across various longer context window sizes at prediction stage. Specifically, we negatively bias attention scores with linearly decreasing penalties that are proportional to query-key distance, which efficiently represents short-term forgetting characteristics of student knowledge states. We conduct comprehensive and rigorous experiments on three real-world educational datasets. The results show that our extraKT model exhibits robust length extrapolation capability and outperforms state-of-the-art baseline models in terms of AUC and accuracy. To encourage reproducible research, we merge our data and code to the publicly available pyKT benchmark at https://github.com/pykt-team/pykt-toolkit.

1 Introduction

Knowledge tracing (KT) is a prediction task that aims to predict students' future performance based on their past learning data, such as their responses to previous exercises, assessments, and engagement with educational content. Figure 1 provides an illustrative example of KT task. Such predictive capabilities can potentially help students learn better and faster when paired with personalized learning materials, which is crucial for developing next-generation intelligent and personalized education [8].

Recently, the rapid progress in attention mechanisms [25] has led to the emergence of various high-performing attention based KT models, such as SAKT [12], SAINT [3], AKT [5] and simpleKT [9]. These KT models effectively extract students' knowledge states by utilizing attention mechanisms to capture the intrinsic relationships between questions and corresponding knowledge components (KCs). The KC is a generalization of everyday terms like concept, principle, fact, or skill. Effectively capturing such relationships may significantly enhance the KT model performance [10, 4].

In online or personalized education settings, students' varying learning paths result in different lengths of student interaction sequences. KT models are expected to handle interaction sequences of varying lengths, which is crucial for real-world educational applications. However, this presents a significant challenge for attention based KT models as the context window sizes in these models are fixed during both the training and prediction stages. To address this problem, we propose enabling the attention based KT models to train on student interactions with a short context window and continue to perform well as the size of the context window increases at the prediction stage. We refer to this as *the length extrapolation of KT model.*

Specifically, we present a novel KT model, namely **extraKT**, which utilizes an efficient position embedding method to facilitate better extrapolation. By negatively biasing attention scores with linearly decreasing penalties proportional to the distance between the relevant key and query, our extraKT is able to learn from student interaction sequences with a short context window and perform well with longer context window sizes at prediction stage. To ensure the reliability and comparability of our results, we choose to follow a publicly available standardized KT task evaluation protocol [8]. We conduct comprehensive and rigorous experiments on three real-world educational datasets, comparing a wide range of baselines. The results demonstrate that our extraKT model exhibits robust length extrapolation capability and outperforms state-of-the-art baseline models in terms of AUC and accuracy.

2 Related Work

2.1 Attention Based KT Models

Attention based KT models capture complex relationships among students' historical interactions by using attention mechanisms. Pandey et al. were the first to utilize attention to predict student knowledge mastery [12]. Since then, employing attention mechanisms to predict students' learning performance has become mainstream. Choi et al. represented students' interactions by using the encoder and decoder of the Transformer [3]. Ghosh et al. proposed a monotonic attention mechanism to model the forgetting behavior of students by introducing an exponential time-related decay [5]. Liu et al. used dot-product attention function to capture question-specific

[*] Corresponding Author: Mingliang Hou. Email: houmingliang@tal.com

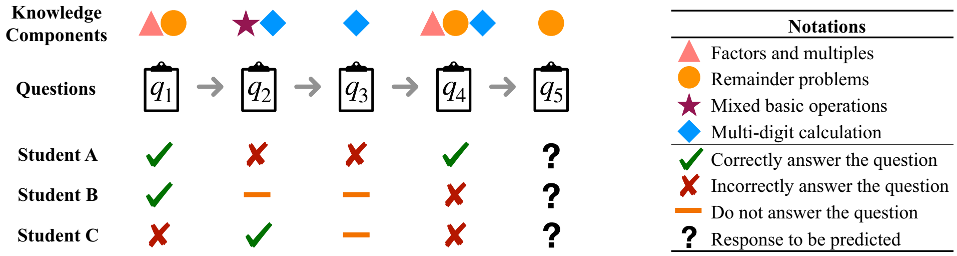

Figure 1: An illustration of the KT task.

variations of the individual differences among questions and their associated KCs [9]. Yin et al. designed a temporal and cumulative attention mechanism to diagnose students' knowledge proficiency from each question mastery state and applied contrastive learning to achieve stable prediction performance [28]. However, in the aforementioned attention based KT models, the context window size has been fixed during training and prediction stages.

2.2 *Length Extrapolation*

The capability of length extrapolation enables a KT model to learn from student interactions with a short context window and continue to perform well as the size of the context window increases during the prediction stage. Position embeddings are crucial for achieving length extrapolation, as demonstrated in various studies [15, 2, 16, 23]. Sinusoidal position embedding combines input embeddings with those from sinusoidal functions, using either fixed or learnable parameters [25, 7]. Unlike Sinusoidal position embedding, Rotary position embedding combines embeddings computed by sinusoidal functions with queries and keys instead of input embeddings [22, 13]. T5 position embedding provides positional information by adding learned biases to attention scores [17]. However, KT models using these position embeddings experience a performance drop during the prediction stage when applied to longer context windows.

Different from existing attention based KT models, inspired by [15, 1], we introduce length extrapolation to make our extraKT model able to learn from student interactions with a short context window and continue to perform well with a longer context window by penalizing attention scores with linear biases.

3 Problem Definition

Given an arbitrary question q^*, a KT model $\mathbf{M}$ aims to predict the probability that a student will correctly answer q^* based on their previous interactions. For each student S, we consider a chronologically ordered sequence of T past interactions, denoted as $S = \{s_i\}_{i=1}^{T}$. Each interaction is represented as a 4-tuple $s =< q, \{c\}, r, t >$, where q denotes the specific question, $\{c\}$ refers to the associated set of KC, r is the binary response indicating whether the student answered the question correctly (1 for correct, 0 for incorrect), and t represents the time step of the response. We would like to estimate the probability p_{q^*} that the student will answer the arbitrary question q^* correctly.

In this paper, our objective is to develop a KT model $\mathbf{M}$ that is able to efficiently extract student knowledge states from student interactions with a short context window and continue to perform well across various longer context window sizes at the prediction stage. We refer to this as *the length extrapolation of KT model*, which is defined as follows:

Definition 1 (Length Extrapolation of KT Model)**.** *Given a student interaction dataset* $\mathbf{D}$*, a KT model* $\mathbf{M}$*, if for any* w_p *that* $w_p > w_t$*, there is,*

$$\frac{|A_p(\mathbf{M},\mathbf{D}) - A_t(\mathbf{M},\mathbf{D})|}{A_t(\mathbf{M},\mathbf{D})} < \epsilon$$

then KT model $\mathbf{M}$ *is considered to have the ability of the length extrapolation, where* A_p *and* A_t *denote the AUC scores on student interactions with context window size* w_p *and* w_t *at prediction and training stage respectively and* ϵ *is a small positive constant.*

4 Our Approach

In this section, we present the framework overview of our extraKT model (as shown in Figure 2), which consists of four components: (1) interaction representation module that encodes questions and responses along with KCs; (2) knowledge extraction module that extracts student knowledge states by capturing relationships between questions and KCs; (3) length extrapolation module that extends the context window to a longer size; and (4) prediction module that uses a two-layer fully connected network to make predictions.

4.1 *Interaction Representation Module*

In real-world educational scenarios, student interaction sequences consist of questions and their associated KCs, as well as corresponding responses. Given that questions and their associated KCs can have intricate relationships and varying levels of difficulty [20, 9], it is crucial to accurately represent student interactions to enhance the performance of KT models. We represent student interactions using a question encoder and response encoder.

4.1.1 *Question Encoder*

Since questions covering the same set of KCs may vary in difficulty levels, students often demonstrate significant differences in performance. To effectively characterize the factor of question difficulty, inspired by the classic and interpretable Rasch model in psychometrics [18, 5, 9], we introduce a learnable question-specific difficulty parameter $\mathbf{d}_{\mathbf{q}_t}$. Intuitively, learning the difficulty parameter based on students' interactions can more effectively and personally model their learning abilities and behaviors. The details of our question encoder are as follows:

$$\mathbf{x}_t = \mathbf{d}_{\mathbf{q}_t} \odot \mathbf{v}_{\mathbf{c}_t} \oplus \mathbf{e}_{\mathbf{c}_t}$$
$$\hat{\mathbf{x}}_t = \mathbf{E}(\mathbf{x}_t)$$

where $\hat{\mathbf{x}}_t$ denotes the latent representation of question $\mathbf{q}_t$ and its associated KC $\mathbf{c}_t$ at the (t)-th time step, obtained through the encoder $\mathbf{E}$, which uses $\mathbf{x}_t$ as the input for queries, keys and values. $\mathbf{d}_{\mathbf{q}_t}$ represents a learnable question difficulty. $\mathbf{v}_{\mathbf{c}_t}$ denotes the KC variation

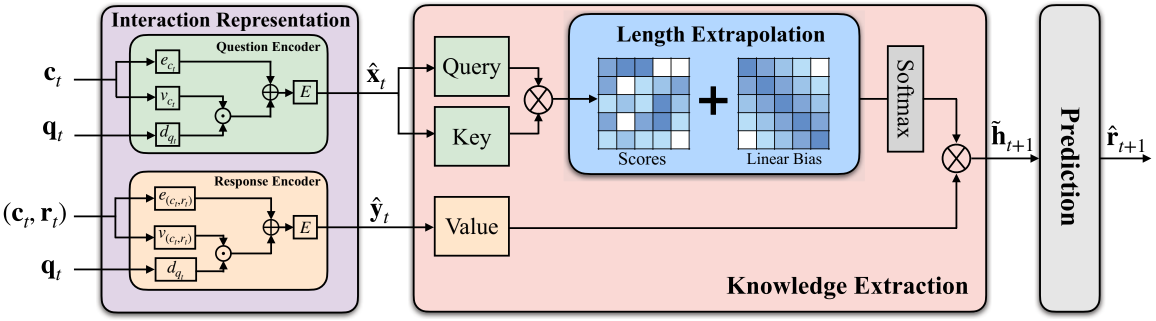

Figure 2: The overview of the proposed extraKT framework.

and $\mathbf{e}_{\mathbf{c}_t}$ denotes the n-dimensional one-hot embeddings of $\mathbf{c}_t$. $\odot$ and $\oplus$ represent the element-wise product and addition operators respectively.

4.1.2 Response Encoder

Based on the question encoder, we extend KC to KC-response pair with a question-specific difficulty parameter to further represent interaction. The details of our response encoder are as following:

$$\mathbf{y}_t = \mathbf{d}_{\mathbf{q}_t} \odot \mathbf{v}_{(\mathbf{c}_t,\mathbf{r}_t)} \oplus \mathbf{e}_{(\mathbf{c}_t,\mathbf{r}_t)}$$
$$\hat{\mathbf{y}}_t = \mathbf{E}(\mathbf{y}_t)$$

where $\hat{\mathbf{y}}_t$ denotes the augmented representation of question $\mathbf{q}_t$ by considering response $\mathbf{r}_t$, obtained through the encoder $\mathbf{E}$, which uses $\mathbf{y}_t$ as the input for queries, keys and values. $\mathbf{d}_{\mathbf{q}_t}$ represents the question difficulty. $\mathbf{e}_{(\mathbf{c}_t,\mathbf{r}_t)}$ denotes the embeddings of $\mathbf{c}_t$ and $\mathbf{r}_t$. $\mathbf{v}_{(\mathbf{c}_t,\mathbf{r}_t)}$ denotes the KC-response variation of $\mathbf{q}_t$ covering this KC $\mathbf{c}_t$ with response $\mathbf{r}_t$. $\odot$ and $\oplus$ represent the element-wise product and addition operators respectively.

4.2 Knowledge Extraction Module

To efficiently explore and extract student knowledge states for better estimation of student knowledge mastery, we choose to use the multi-head attention mechanism with dot-product attention, which differs from many existing KT methods based on sequential neural networks. The dot-product attention can extract the time-aware and contextual information embedded in the student interactions and the multi-head attention mechanism can capture more intricate features. Specifically, the extracted knowledge state $\mathbf{h}_{t+1}$ at the (t+1)-th time step is calculated as follows:

$$\mathbf{Q} = \hat{\mathbf{x}}_{t+1}; \mathbf{K} = \{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_t\}; \mathbf{V} = \{\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_t\}$$
$$\text{Head} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}) \cdot \mathbf{V}$$
$$\mathbf{h}_{t+1} = \text{Concat}(\text{Head}_1, \text{Head}_2, \cdots, \text{Head}_n)$$

where $\mathbf{K}^T$ denotes the transpose of $\mathbf{K}$ and d is the dimension of $\mathbf{K}$. Head_n is the (n)-th head of multi-head attention.

4.3 Length Extrapolation Module

To facilitate better extrapolation of our extraKT model, inspired by [15], we choose to utilize an efficient position embedding method that biases attention scores with proportional penalties based on the distance between queries and keys. Specifically, the linear biases enable the model to adapt to various context window sizes by providing relative position information of student interactions. Furthermore, we propose that attention scores with linear biases are able to efficiently represent short-term forgetting characteristics of student knowledge states. By introducing length extrapolation, the extracted knowledge state $\mathbf{h}_{t+1}$ can be reformulated as follows:

$$\tilde{\text{Head}} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{B}) \cdot \mathbf{V}$$
$$\tilde{\mathbf{h}}_{t+1} = \text{Concat}(\tilde{\text{Head}}_1, \tilde{\text{Head}}_2, \cdots, \tilde{\text{Head}}_i)$$

where $\tilde{\mathbf{h}}_{t+1}$ denotes the extracted knowledge states with length extrapolation and $\mathbf{B}$ represents the matrix of linear biases. Specifically, each element of $\mathbf{B}$ is calculated by:

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} \\ b_{21} & b_{22} & \cdots & b_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ij} \end{bmatrix}$$
$$b_{ij} = -\beta \cdot |i - j|$$

where b_{ij} denotes the element at the (i)-th row and the (j)-th column of $\mathbf{B}$. β is the slope coefficient that adjusts the attention scores for the (n)-th attention head out of H attention heads. In here, we set β to $2^{-8\frac{n}{H}}$.

4.4 Prediction Module

In this section, we utilize a two-layer fully connected neural network to predict student responses. To optimize the prediction function, we minimize the binary cross-entropy loss between the student's ground-truth response $\mathbf{r}_{t+1}$ and the predicted response $\hat{\mathbf{r}}_{t+1}$ [29, 24]. This approach ensures that our model learns to accurately estimate the probability of a student answering correctly, thereby improving its predictive performance.

$$\hat{\mathbf{r}}_{t+1} = \sigma(\text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \cdot [\tilde{\mathbf{h}}_{t+1}; \hat{\mathbf{x}}_{t+1}] + \mathbf{k}_1) + \mathbf{k}_2))$$
$$\mathcal{L} = -\sum_t (\mathbf{r}_{t+1} \cdot \log \hat{\mathbf{r}}_{t+1} + (1 - \mathbf{r}_{t+1}) \cdot \log(1 - \hat{\mathbf{r}}_{t+1}))$$

where σ denotes Sigmoid function. $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{W}_1$, $\mathbf{W}_2$ are trainable parameters and $\mathcal{L}$ represents the binary cross-entropy loss function.

5 Experiments

5.1 Datasets

We select three widely used benchmark datasets to evaluate the effectiveness of our model as follows:

Table 1: Data statistics of three widely used datasets.

	# of interactions	**# of students**	**# of questions**	**# of KCs**	**Avg. interactions per student**	**Percentage of length $\geqslant$ 200**
AL2005	607,021	574	173,113	112	1,057.5	81.71%
BD2006	1,817,458	1,145	129,263	493	1,587.3	92.75%
NIPS34	1,382,678	4,918	948	57	281.1	58.72%

- Algebra 2005-2006 (AL2005): This dataset originates from KDD Cup 2010 EDM Challenge which includes 13-14 year-old students' interactions with Algebra questions. It has detailed step-level student responses to the mathematical problems [21]. In our experiment, we use the concatenation of the problem name and step name as a unique question.
- Bridge to Algebra 2006-2007 (BD2006): The BD2006 dataset consists of mathematical problems extracted from logs of students' interactions with intelligent tutoring systems [21]. The unique question construction in BD2006 follows a format similar to AL2005.
- NeurIPS2020 Education Challenge (NIPS34): This dataset is provided by NeurlPS 2020 Education Challenge. We use the dataset of Task 3 & Task 4 to evaluate our models [26]. It includes students' responses to mathematics questions from Eedi, a platform with millions of daily student interactions worldwide. We opt to use the leaf nodes from the subject tree as its KCs.

Please note that there are many datesets available in knowledge tracing, such as AL2005, BD2006, NIPS34, Statics2011, AS2009, AS2015 and POJ [9]. However, for datasets to better reflect real-world educational scenarios, they need to include both questions and their associated KCs. Among the aforementioned datasets, only AL2005, BD2006, NIPS34, and AS2009 fulfill this criterion. In order to study the impact of long content window in attention based KT models, it is important to have a sufficient amount of long-sequence data, where sequences exceeding a length of 200 account for over 50%. This requirement is only satisfied by AL2005, BD2006, and NIPS34. Therefore, we have chosen these three datasets for our research. To ensure reproducibility in our experiments, we rigorously follow the data pre-processing steps suggested in [8]. Data statistics are summarized in Table 1.

5.2 *Baselines*

We compare our extraKT model with the following state-of-the-art KT models to evaluate the effectiveness of our approach:

- DKT [14]: It is the first model to incorporate deep learning into the KT task. Specifically, it uses recurrent neural networks (RNNs) to model student learning processes and to estimate the student mastery of questions and corresponding KCs.
- DKT+ [27]: This method aims to address two key challenges encountered in DKT. First, DKT struggles with accurately reconstructing the observed input. Second, there is inconsistency in the performance of KCs across different time steps. It employs both L1-norm and L2-norm to quantify the disparity between two adjacent prediction results.
- DKVMN [30]: This memory-augmented neural network utilizes a key matrix to capture the relationships among underlying KCs and employs a value matrix to denote the student's proficiency level for each KC at every time step.
- GKT [11]: Inspired by the graph-like structure in educational coursework, it represents the knowledge hierarchy as a graph and transforms the KT task into a time-series node-level classification challenge within graph neural networks (GNNs). Additionally, the authors introduce several strategies to address the absence of explicit graph structures in numerous datasets.
- LPKT [19]: It estimates students' knowledge states through directly modeling their learning processes. Specifically, it uses a learning gate to distinguish students' absorptive capacity of knowledge and forgetting gate to model the decline of students' knowledge over time.
- SAKT [12]: To address the generalization issues associated with sparse data encountered in other models, this approach leverages a self-attention mechanism to capture the relationships between questions and KCs. It employs question embeddings as queries and utilizes interaction embeddings as both keys and values for the attention mechanism.
- SAINT [3]: It employs a Transformer based architecture for KT task, where the encoder utilizes self-attention to process the sequence of student interactions, while the decoder employs self-attention and masked encoder-decoder attention to handle the sequence of student responses.
- AKT [5]: It is an attention based model that integrates a novel monotonic attention mechanism to link students' future performance with their past responses. Additionally, it employs a Rasch model to regularize the questions and KCs embeddings.
- ATKT [6]: This model, with an attention-LSTM backbone, uses adversarial perturbations to enhance the generalization of KT models and mitigate the overfitting issue commonly encountered in deep neural network (DNN) based KT models. Adversarial perturbations, in combination with the original interaction embeddings, contribute to predicting students' performance.
- simpleKT [9]: It uses scaled dot-product attention mechanism to capture complex relationships between questions and corresponding KCs. To capture the individual differences among questions on the same KC, it defines a question-specific difficulty vector.
- DTransformer [28]: This approach utilizes a Transformer based model within a two-level framework to achieve knowledge state estimation while ensuring stability through contrastive learning.

5.3 *Experimental Setting*

We train all models on student interactions with a fixed context window size of 200 and evaluate them on sequences with context window sizes of 200, 400, 600, 800 and 1000, respectively. For each combination of models and datasets, we perform standard 5-fold cross-validation. We exclude interactions that lack a student ID or any required information from our 4-tuple interaction representation and filter out students with fewer than three interactions. For the test set, we randomly withhold 20% of students and their interaction sequences. The remaining 80% of students are randomly and evenly split into 5 folds, with 4 folds used for training and 1 fold for validation. We implement early stopping if there is no improvement in performance after 10 epochs. The Adam optimizer is used to train the models for up to 200 epochs for each hyperparameter combination,

Table 2: AUC and accuracy performance comparisons on AL2005 dataset.

Model	AUC					Accuracy				
	Size of Context Window					Size of Context Window				
	200	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
DKT	0.8149±0.0011	0.8150±0.0011	0.8150±0.0011	0.8149±0.0011	0.8149±0.0011	0.8097±0.0005	0.8098±0.0005	0.8098±0.0006	0.8098±0.0006	0.8098±0.0006
DKT+	0.8156±0.0011	0.8156±0.0010	0.8156±0.0010	0.8156±0.0010	0.8156±0.0010	0.8097±0.0007	0.8098±0.0006	0.8098±0.0006	0.8097±0.0006	0.8097±0.0006
DKVMN	0.8054±0.0011	0.8039±0.0014	0.8030±0.0016	0.8025±0.0017	0.8023±0.0018	0.8027±0.0007	0.8025±0.0008	0.8023±0.0008	0.8022±0.0008	0.8022±0.0009
GKT	0.8110±0.0009	0.8111±0.0009	0.8111±0.0009	0.8111±0.0009	0.8111±0.0009	0.8088±0.0008	0.8087±0.0010	0.8088±0.0010	0.8088±0.0010	0.8088±0.0010
LPKT	0.8268±0.0004	0.8216±0.0019	0.8107±0.0104	0.7990±0.0181	0.7891±0.0197	**0.8154±0.0008**	**0.8123±0.0017**	0.7970±0.0217	0.7746±0.0543	0.7613±0.0694
SAKT	0.7899±0.0036	0.6743±0.0023	0.6691±0.0030	0.6677±0.0024	0.6666±0.0018	0.7965±0.0019	0.7478±0.0016	0.7468±0.0026	0.7445±0.0017	0.7435±0.0020
SAINT	0.7715±0.0018	0.6691±0.0110	0.6589±0.0021	0.6539±0.0017	0.6551±0.0016	0.7755±0.0012	0.7355±0.0118	0.7424±0.0050	0.7291±0.0092	0.7324±0.0108
AKT	0.8306±0.0013	0.8277±0.0030	0.8258±0.0038	0.8241±0.0045	0.8227±0.0051	0.8124±0.0011	0.8117±0.0011	**0.8108±0.0013**	0.8100±0.0018	0.8094±0.0023
ATKT	0.7995±0.0023	0.7816±0.0025	0.7641±0.0039	0.7523±0.0047	0.7446±0.0050	0.7998±0.0019	0.7935±0.0026	0.7854±0.0049	0.7779±0.0072	0.7731±0.0090
simpleKT	0.8210±0.0014	0.7808±0.0078	0.7763±0.0055	0.7535±0.0263	0.7655±0.0169	0.8067±0.0011	0.7921±0.0026	0.7899±0.0031	0.7820±0.0042	0.7877±0.0032
DTransformer	0.8188±0.0025	0.8156±0.0025	0.8137±0.0028	0.8123±0.0030	0.8112±0.0033	0.8043±0.0021	0.8032±0.0021	0.8023±0.0023	0.8018±0.0023	0.8013±0.0026
extraKT	**0.8317±0.0021**	**0.8317±0.0020**	**0.8317±0.0019**	**0.8317±0.0019**	**0.8317±0.0019**	0.8110±0.0009	0.8109±0.0010	**0.8108±0.0011**	**0.8108±0.0010**	**0.8109±0.0011**

Table 3: AUC and accuracy performance comparisons on BD2006 dataset.

Model	AUC					Accuracy				
	Size of Context Window					Size of Context Window				
	200	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
DKT	0.8015±0.0008	0.8015±0.0008	0.8015±0.0008	0.8015±0.0008	0.8015±0.0008	0.8553±0.0002	0.8553±0.0002	0.8552±0.0002	0.8552±0.0002	0.8552±0.0002
DKT+	0.8020±0.0004	0.8021±0.0004	0.8021±0.0004	0.8021±0.0004	0.8021±0.0004	0.8553±0.0003	0.8553±0.0003	0.8553±0.0003	0.8553±0.0003	0.8553±0.0003
DKVMN	0.7983±0.0009	0.7956±0.0009	0.7936±0.0010	0.7925±0.0012	0.7919±0.0014	0.8545±0.0002	0.8540±0.0003	0.8537±0.0002	0.8535±0.0001	0.8534±0.0001
GKT	0.8046±0.0008	0.8047±0.0009	0.8047±0.0009	0.8047±0.0010	0.8047±0.0010	0.8511±0.0004	0.8555±0.0002	0.8556±0.0002	0.8556±0.0002	0.8556±0.0002
LPKT	0.8056±0.0008	0.8014±0.0021	0.7965±0.0029	0.7939±0.0031	0.7923±0.0031	0.8547±0.0005	0.8539±0.0004	0.8524±0.0009	0.8507±0.0021	0.8495±0.0032
SAKT	0.7739±0.0015	0.7097±0.0056	0.7000±0.0042	0.6987±0.0035	0.6962±0.0044	0.8460±0.0004	0.8190±0.0030	0.8208±0.0030	0.8240±0.0008	0.8239±0.0009
SAINT	0.7791±0.0018	0.6847±0.0035	0.6816±0.0027	0.6692±0.0037	0.6697±0.0024	0.8445±0.0013	0.8396±0.0006	0.8373±0.0014	0.8396±0.0006	0.8396±0.0006
AKT	0.8208±0.0007	0.8187±0.0008	0.8168±0.0010	0.8155±0.0012	0.8144±0.0014	0.8587±0.0005	0.8581±0.0004	0.8575±0.0005	0.8571±0.0004	0.8567±0.0005
ATKT	0.7889±0.0008	0.7641±0.0028	0.7370±0.0041	0.7142±0.0042	0.6963±0.0040	0.8555±0.0002	0.8432±0.0020	0.8334±0.0033	0.8241±0.0043	0.8156±0.0058
simpleKT	0.8151±0.0006	0.7897±0.0046	0.7764±0.0124	0.7726±0.0090	0.7724±0.0088	0.8567±0.0010	0.8506±0.0011	0.8444±0.0059	0.8484±0.0024	0.8434±0.0049
DTransformer	0.8093±0.0009	0.8052±0.0020	0.8023±0.0029	0.8002±0.0035	0.7985±0.0039	0.8555±0.0007	0.8544±0.0007	0.8539±0.0010	0.8532±0.0010	0.8529±0.0010
extraKT	**0.8247±0.0006**	**0.8246±0.0005**	**0.8246±0.0005**	**0.8245±0.0005**	**0.8245±0.0005**	**0.8605±0.0012**	**0.8605±0.0011**	**0.8605±0.0011**	**0.8605±0.0011**	**0.8605±0.0011**

Table 4: AUC and accuracy performance comparisons on NIPS34 dataset.

Model	AUC					Accuracy				
	Size of Context Window					Size of Context Window				
	200	**400**	**600**	**800**	**1000**	**200**	**400**	**600**	**800**	**1000**
DKT	0.7689±0.0002	0.7689±0.0002	0.7689±0.0002	0.7689±0.0002	0.7689±0.0002	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004
DKT+	0.7696±0.0002	0.7697±0.0002	0.7697±0.0002	0.7697±0.0002	0.7697±0.0002	0.7039±0.0004	0.7039±0.0004	0.7039±0.0004	0.7039±0.0004	0.7039±0.0004
DKVMN	0.7673±0.0004	0.7673±0.0004	0.7673±0.0004	0.7672±0.0004	0.7672±0.0004	0.7016±0.0005	0.7015±0.0005	0.7015±0.0005	0.7015±0.0005	0.7015±0.0005
GKT	0.7689±0.0024	0.7689±0.0025	0.7689±0.0025	0.7689±0.0025	0.7689±0.0025	0.7014±0.0028	0.7013±0.0029	0.7013±0.0029	0.7013±0.0029	0.7013±0.0029
LPKT	0.8004±0.0003	0.7997±0.0005	0.7993±0.0006	0.7992±0.0007	0.7992±0.0006	0.7309±0.0006	0.7303±0.0012	0.7298±0.0015	0.7297±0.0016	0.7297±0.0015
SAKT	0.7525±0.0009	0.7331±0.0013	0.7329±0.0011	0.7330±0.0011	0.7330±0.0011	0.6884±0.0009	0.6741±0.0012	0.6739±0.0009	0.6740±0.0010	0.6740±0.0010
SAINT	0.7895±0.0009	0.7708±0.0009	0.7703±0.0012	0.7700±0.0012	0.7700±0.0012	0.7204±0.0009	0.7029±0.0012	0.7024±0.0012	0.7021±0.0012	0.7021±0.0012
AKT	0.8033±0.0003	0.8030±0.0004	0.8028±0.0004	0.8028±0.0004	0.8028±0.0004	0.7323±0.0005	0.7319±0.0006	0.7318±0.0005	0.7318±0.0005	0.7318±0.0005
ATKT	0.7665±0.0001	0.7630±0.0005	0.7620±0.0006	0.7619±0.0006	0.7619±0.0006	0.7013±0.0002	0.6988±0.0005	0.6980±0.0008	0.6980±0.0007	0.6980±0.0007
simpleKT	0.8035±0.0000	0.7952±0.0017	0.7961±0.0012	0.7960±0.0012	0.7960±0.0012	0.7328±0.0001	0.7251±0.0016	0.7260±0.0013	0.7259±0.0013	0.7259±0.0013
DTransformer	0.7994±0.0003	0.7988±0.0003	0.7985±0.0003	0.7985±0.0003	0.7985±0.0003	0.7295±0.0007	0.7289±0.0006	0.7286±0.0007	0.7286±0.0007	0.7286±0.0007
extraKT	**0.8045±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.7340±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**

and we employ Bayesian search to identify the optimal hyperparameters for each fold. We set the embedding dimension, hidden state dimension, and prediction layer dimension to [64, 128, 256]. The learning rate, dropout rate, and random seed are set to [1e-3, 1e-4, 1e-5], [0.05, 0.1, 0.3, 0.5] and [42, 3407], respectively. Consistent with previous studies [14, 5, 8], we report the average AUC and accuracy, as well as the standard deviations across 5 folds to evaluate the KT prediction performance.

5.4 Results

5.4.1 Overall Performance

Tables 2 - 4 show the overall performance. The best AUC and accuracy are in **bold** and the second-best AUC and accuracy are underlined. From these tables, we have the following observations: (1) Our extraKT model outperforms almost all state-of-the-art models (except LPKT in terms of the accuracy performance metric at the context window sizes of 200 and 400 on the AL2005 dataset) and maintains stable performance as the context window size increases on all three datasets. This suggests that we effectively extend the context window of our model by length extrapolation, enabling it to better and consistently extract the knowledge states of students even as the context window size varies. (2) On NIPS34 dataset (in Table 4), compared to the other two datasets, some attention based models (such as SAKT, SAINT and AKT etc.) do not exhibit significant drops in performance at context window sizes of 600, 800 and 1000 in terms of AUC and accuracy. This is because the average number of interactions per student in NIPS34 dataset is only 281.1 (as shown in Table 1), compared to the other two datasets, AL2005 and BD2006, which have averages of 1,057.5 and 1,587.3 respectively. When the number of student interactions is less than context window, extending context window size has little impact on the prediction performance. (3) Some models (such as DKT, DKT+ and GKT etc.) do not experience notable drops in performance as the context window size increases on all three datasets. Since these models do not rely on attention mechanisms, they typically do not face length extrapolation challenges. However, these models exhibit significantly lower performance, compared to our attention based KT model, extraKT. For example, compared to DKT, our extraKT has a large improvement of

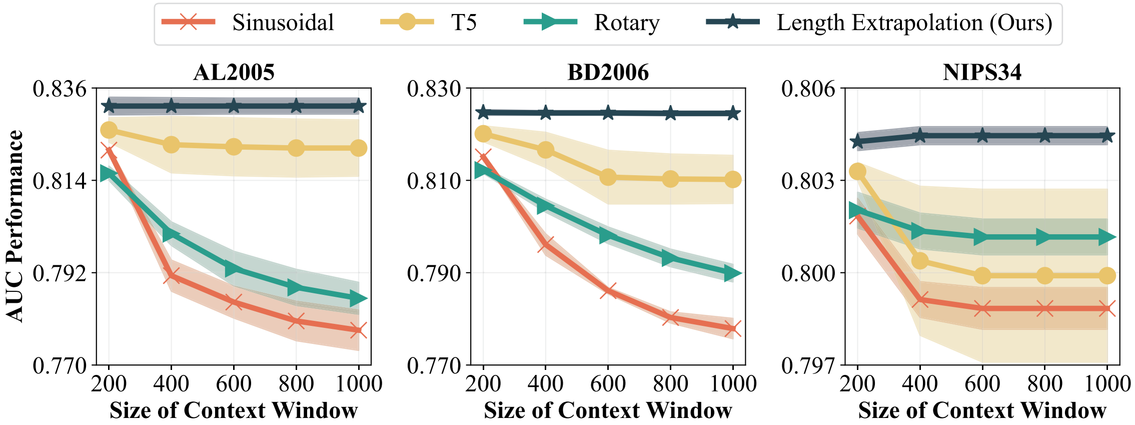

Figure 3: Analysis with different position embeddings in AUC performance.

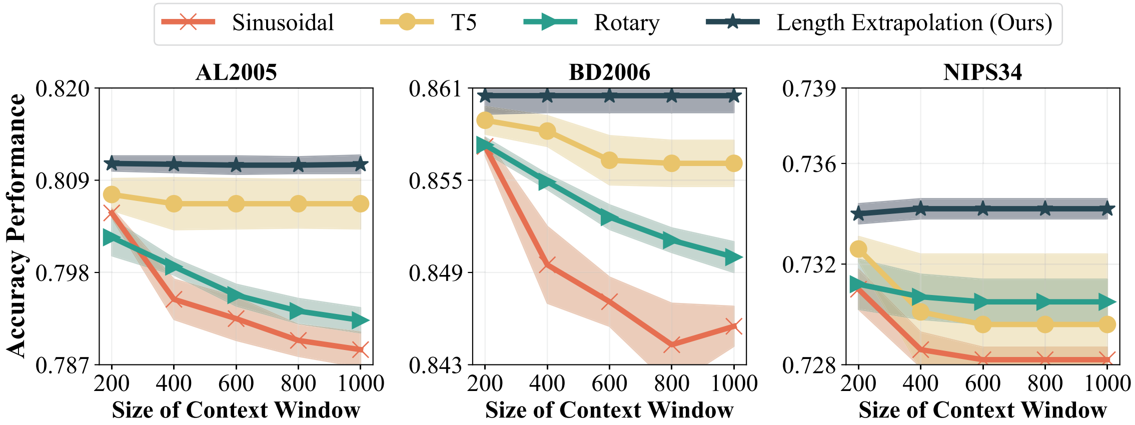

Figure 4: Analysis with different position embeddings in accuracy performance.

AUC by 1.68%, 2.32% and 3.56% at the context window size of 200 on the three datasets.

Please note that although the modest AUC increase in Tables 2 - 4 is less than 1% compared to the best baseline at the context window size of 200, this improvement is significant. Recent benchmark research reveals that many reported performance gains are unreliable due to the reckless evaluation process and there has been only a 3.5% enhancement in overall KT prediction performance since 2015. In our study, we strictly follow the evaluation process proposed by Liu et al. [8] and conduct comprehensive hyperparameter search for every baseline. We have made all details of our approach available at https://github.com/pykt-team/pykt-toolkit to ensure our results are reliable and reproducible.

5.4.2 *Impact on Different Position Embeddings*

To further explore the impact on different position embeddings, we conduct experiments on extraKT model with Sinusoidal, T5, Rotary and our length extrapolation module respectively. Figures 3 - 4 show the results. From these figures, we have the following observations: (1) Different position embeddings significantly influence the length extrapolation of KT model. On AL2005 and BD2006 datasets, both Sinusoidal and Rotary exhibit notable decreases in terms of AUC and accuracy as the context window size increases, while our length extrapolation module maintains more stable performance. However, when evaluating on the NIPS34 dataset, Rotary outperforms T5, yet our length extrapolation module consistently maintains stable performance. This indicates that Rotary is better suited for scenarios with fewer interactions per student compared to T5. In contrast, our length extrapolation module effectively accommodating both shorter and longer sequences per student. (2) Different position embeddings affect prediction performance of KT model. Our length extrapolation module performs the best, followed by T5 or Rotary, while the Sinusoidal performs the worst on the three datasets. This is because, compared to other position embeddings, our length extrapolation module provides position information by linear biases in a computationally friendly way, which prevents the model overfit on position information of student interaction sequences and efficiently represent short-term forgetting characteristics of student knowledge states.

5.4.3 *Ablation Study*

We conduct ablation experiments to show how length extrapolation module in our extraKT model affects the performances in Figures 5 - 6. The LE represents the length extrapolation module and the w/o means excluding such module from extraKT. From these figures, we observe that the extraKT model exhibits a notable performance drop without length extrapolation module on all three datasets. This suggests that we effectively extend context window of our extraKT via length extrapolation.

5.4.4 *Visualization*

To visualize the impact of length extrapolation on attention, we compare the original attention scores with those enhanced by length ex-

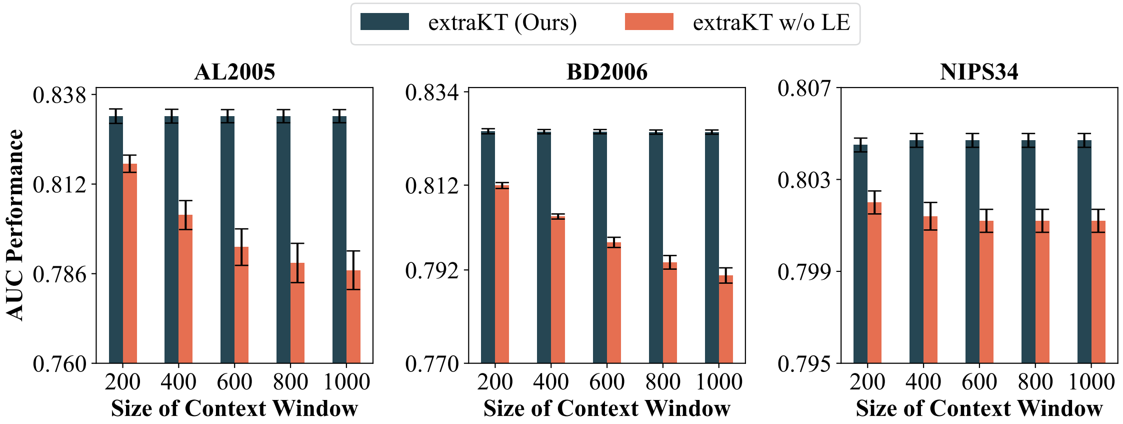

Figure 5: Component analysis of extraKT on three datasets in AUC performance.

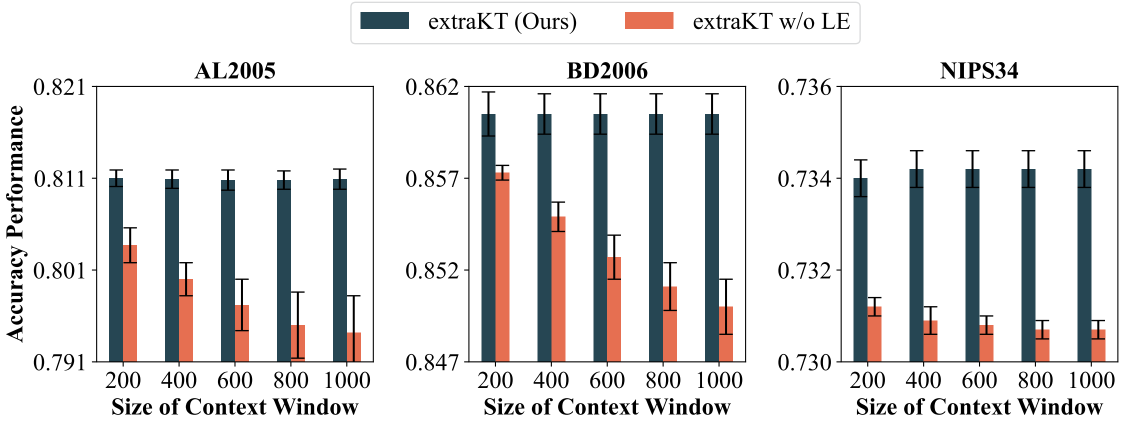

Figure 6: Component analysis of extraKT on three datasets in accuracy performance.

trapolation, as shown in Figure 7. From Figure 7, we observe that original attention with length extrapolation can efficiently represent student short-term forgetting characteristics, while original attention tends to overfit on position information of student interaction sequences. Specifically, when calculating similarities between questions, the original attention mostly focuses on question index 1, while original attention with length extrapolation gradually shifts towards allocating attention to recent questions over time. This indicates that our length extrapolation module is well-designed for modeling students' short-term forgetting behaviors.

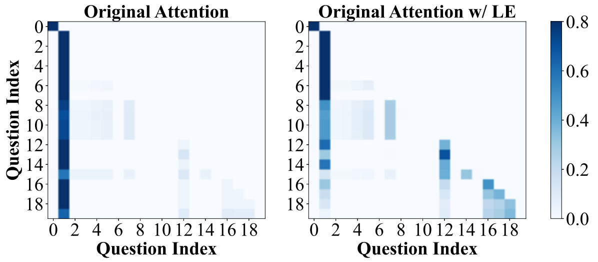

Figure 7: Visualization of length extrapolation (LE) impact on attention.

6 Conclusion and Future Work

In this paper, we propose extraKT model, designed to facilitate better extrapolation. This model learns from student interactions with short context window and continue to perform well as the size of context window increases at prediction stage. Compared with existing KT models, our extraKT model effectively represents student short-term forgetting characteristics of knowledge states. Experimental results on three real-world educational datasets demonstrate that our extraKT model exhibits robust length extrapolation capability and outperforms state-of-the-art baseline models in terms of AUC and accuracy.

In the future, there are some points that we need further discuss and study: (1) When considering lifelong learning, the length of student interaction sequences continuously grows. However, on existing real-world educational datasets, such as the AL2005 and BD2006 datasets, the average sequence length per student is only slightly over one thousand. How to bridge the gap between real-world scenarios and datasets when evaluating length extrapolation capability of KT models remains a challenge. (2) The length extrapolation problem also arises in other educational scenarios, such as adaptive testing systems, student writing analysis, and so on, which require models to handle continuously growing data effectively. Our method has the potential to be adapted and applied to these tasks.

Acknowledgements

This work was supported in part by National Key R&D Program of China, under Grant No. 2022YFC3303600 and in part by Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University (2022LSYS003).

References

[1] T.-C. Chi, T.-H. Fan, P. J. Ramadge, and A. Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, New Orleans, Louisiana, USA, December 2022.

[2] T.-C. Chi, T.-H. Fan, A. Rudnicky, and P. Ramadge. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, July 2023.

[3] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the 7th ACM Conference on Learning at Scale*, Virtual Conference, August 2020.

[4] J. Cui, Z. Chen, A. Zhou, J. Wang, and W. Zhang. Fine-grained interaction modeling with multi-relational transformer for knowledge tracing. *ACM Transactions on Information Systems*, **41**:1–26, January 2023.

[5] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Virtual Conference, August 2020.

[6] X. Guo, Z. Huang, J. Gao, M. Shang, M. Shu, and J. Sun. Enhancing knowledge tracing via adversarial training. In *Proceedings of the 29th ACM International Conference on Multimedia*, Virtual Conference, October 2021.

[7] D. Jacob, C. Ming-Wei, L. Kenton, and T. Kristina. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, June 2019.

[8] Z. Liu, Q. Liu, J. Chen, S. Huang, J. Tang, and W. Luo. pykt: A python library to benchmark deep learning based knowledge tracing models. In *Proceedings of 36th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, New Orleans, LA, USA, November 2022.

[9] Z. Liu, Q. Liu, J. Chen, S. Huang, and W. Luo. simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda, May 2023.

[10] Z. Liu, Q. Liu, T. Guo, J. Chen, S. Huang, X. Zhao, J. Tang, W. Luo, and J. Weng. Xes3g5m: a knowledge tracing benchmark dataset with auxiliary information. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, December 2023.

[11] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence*, Thessaloniki, Greece, October 2019.

[12] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, Montréal, Canada, July 2019.

[13] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. YaRN: Efficient context window extension of large language models. In *Proceedings of the 12th International Conference on Learning Representations*, Vienna Austria, May 2024.

[14] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th Conference on Neural Information Processing Systems*, Montreal, Quebec, Canada, December 2015.

[15] O. Press, N. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *Proceedings of the 10th International Conference on Learning Representations*, Virtual Conference, April 2022.

[16] Z. Qin, Y. Zhong, and H. Deng. Exploring transformer extrapolation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, February 2024.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, **21**(140):1–67, February 2020.

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

[19] S. Shen, Q. Liu, E. Chen, Z. Huang, W. Huang, Y. Yin, Y. Su, and S. Wang. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Conference, August 2021.

[20] S. Shen, Z. Huang, Q. Liu, Y. Su, S. Wang, and E. 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*, New York, USA, July 2022.

[21] J. Stamper and Z. A. Pardos. The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics. *Journal of Learning Analytics*, **3**(2):312–316, 2016.

[22] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, **568**:127063, February 2024.

[23] Y. Sun, L. Dong, B. Patra, S. Ma, S. Huang, A. Benhaim, V. Chaudhary, X. Song, and F. Wei. A length-extrapolatable transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, July 2023.

[24] R. Vashisht and H. G. Ramaswamy. On the learning dynamics of attention networks. In *Proceedings of the 26th European Conference on Artificial Intelligence*, Kraków, Poland, October 2023.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, December 2017.

[26] Z. Wang, A. Lamb, E. Saveliev, P. Cameron, Y. Zaykov, J. M. Hernández-Lobato, R. E. Turner, R. G. Baraniuk, C. Barton, S. P. Jones, et al. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*, 2020.

[27] C.-K. Yeung and D.-Y. Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the 5th Annual ACM Conference on Learning at Scale*, London, UK, June 2018.

[28] Y. Yin, L. Dai, Z. Huang, S. Shen, F. Wang, Q. Liu, E. Chen, and X. Li. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the ACM Web Conference 2023*, Austin, TX, USA, April 2023.

[29] B. Zhan, T. Guo, X. Li, M. Hou, Q. Liang, B. Gao, W. Luo, and Z. Liu. Knowledge tracing as language processing: A large-scale autoregressive paradigm. In *Proceedings of the 25th International Conference on Artificial Intelligence in Education*, Recife, Brazil, July 2024.

[30] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, Perth, Canada, April 2017.