Cognitive Fluctuations Enhanced Attention Network for Knowledge Tracing

Mingliang Hou[1, 2], **Xueyi Li**[1], **Teng Guo**[*1], **Zitao Liu**[1], **Mi Tian**[2], **Renqiang Luo**[3], **Weiqi Luo**[1]

[1]Guangdong Institute of Smart Education, Jinan University, Guangdong, 510632, China
[2]TAL Education Group, Beijing, 102206, China
[3]School of Software Technology, Dalian University of Technology, Liaoning, 116622, China
teemohold@outlook.com, lixueyi@stu2021.jnu.edu.cn, {guoteng, liuzitao}@jun.edu.cn, tinami@tal.com, lrenqiang@outlook.com, lwq@jnu.edu.cn

Abstract

Knowledge tracing (KT) involves using the historical records of student-learning interactions to anticipate their performance on forthcoming questions. Central to this process is the modeling of human cognition to gain deeper insights into how knowledge is acquired and retained. Human cognition is characterized by two key features: long-term cognitive trends, reflecting the gradual accumulation and stabilization of knowledge over time, and short-term cognitive fluctuations, which arise from transient factors such as forgetting or momentary lapses in attention. Although existing attention-based KT models effectively capture long-term cognitive trends, they often fail to adequately address short-term cognitive fluctuations. These limitations lead to overly smoothed cognitive features and reduced model performance, especially when the test data length exceeds the training data length. To address these problems, we propose FlucKT, a novel short-term cognitive fluctuations enhanced attention network for KT tasks. FlucKT improves the attention mechanism in two ways: First, by using a decomposition-based layer with causal convolution to separate and dynamically reweight long-term and short-term cognitive features. Second, by introducing a kernelized bias attention score penalty to enhance focus on short-term fluctuations, improving length generalization capabilities. Our contributions are validated through extensive experiments on three real-world datasets, demonstrating significant improvements in length generalization and prediction performance.

Code — https://pykt.org/
Extended version — https://pykt.org/

Introduction

Knowledge tracing (KT) is a sequential prediction task that leverages students' historical learning interaction data to forecast their performance on future questions. The essence of KT lies in modeling human cognitive behaviors to deepen understanding of cognitive processes. Consequently, addressing the KT task enables teachers to more effectively guide students who need additional support and to recommend personalized learning materials, which is crucial for advancing next-generation intelligent and personalized education. To accurately capture the dynamic information of students, significant efforts in recent years have employed either Markov chains (Yudelson, Koedinger, and Gordon 2013) or recurrent neural networks (RNNs) (Piech et al. 2015; Liu et al. 2023). Meanwhile, the Transformer architecture (Vaswani et al. 2017) has gained prominence for surpassing RNN-based models in KT tasks due to its ability to model long-range dependencies. As a result, various KT models (Ghosh, Heffernan, and Lan 2020; Liu et al. 2022a; Im et al. 2023; Yin et al. 2023) have adopted the Transformer as the sequence encoder to capture correlations in knowledge states by assigning attention weights to different positions, thereby achieving high-quality sequence representations.

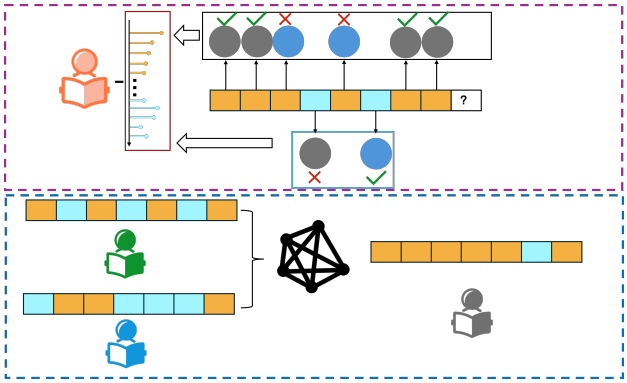


Figure 1: The purple dashed box highlights their dynamic interaction within a student's learning sequence, while the blue dashed box showcases how repeated attention layers aggregate these features across multiple students, blurring the distinction between the two types of cognition. Different colored solid circles represent distinct KCs. The black and blue solid boxes indicate long-term cognitive trends and short-term cognitive fluctuations, respectively. The red solid box represents the frequency domain of sequence embedding. Solid square blocks denote features.

Although attention mechanisms excel in capturing long-term cognitive trends in KT sequence data, they often do not adequately address short-term cognitive fluctuations. This

*Corresponding author.

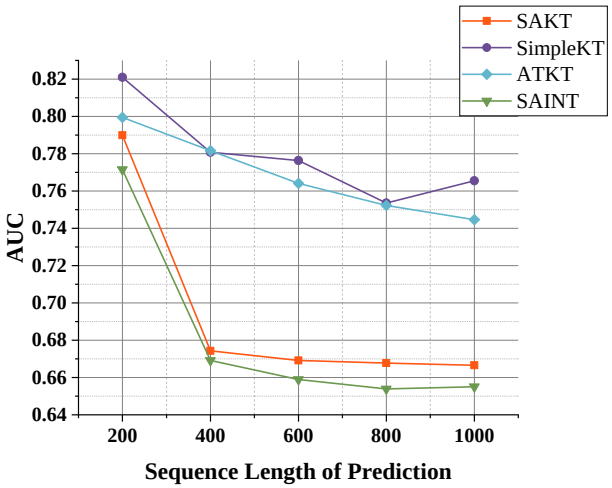


Figure 2: Comparison of AUC performance for four KT models across different sequence lengths.

leads to two key limitations in existing attention-based KT models: 1) In KT, long-term cognitive trends generally reflect the gradual accumulation and stabilization of a student's knowledge state over time. In contrast, short-term cognitive fluctuations are more transient and often arise from random cognitive factors such as forgetting, fatigue-induced lapses in attention, carelessness, or guessing (Salthouse, Nesselroade, and Berish 2006; Mozer et al. 2009). Long-term cognitive trends and short-term cognitive fluctuations influence the overall knowledge state of the students, as illustrated in the upper part (red dashed box) of Fig. 1. Existing attention mechanisms may inadvertently filter out cognitive patterns associated with short-term fluctuations, leading to overly smoothed cognitive behavior features that lack distinctiveness. This impairs the model's ability to accurately capture and represent student cognitive behaviors, as shown in the lower part (blue dashed box) of Fig. 1. 2) Excessive focus on long-term cognitive trends in KT sequences can cause the attention score matrix to become too diffuse. When the length of the inference or test data exceeds the length of the training data (i.e., train short, test long), the distribution of attention scores may become too spread out, resulting in decreased model performance, as depicted in Fig. 2. In real-world online learning systems, the length of students' interaction data is variable. An ideal KT model should exhibit robust length generalization capabilities, meaning that it should be trained with shorter context windows and continue to perform well as the context window size increases during the prediction phase.

To address these two limitations, we propose a short-term cognitive fluctuations enhanced attention network for knowledge tracing, called FlucKT. Specifically, FlucKT improves the current attention mechanism in two ways: First, we design a decomposition-based layer to enhance the input features of the attention mechanism. This design uses causal convolution to decompose attention input into long-term cognitive trends and short-term cognitive fluctuations, and then dynamically reweights and recombines them. This enables the attention mechanism to adaptively determine the focus on persistent cognitive states and transient cognitive states. Second, we design a kernelized bias attention score penalty mechanism-based on the distance between the query and the key to enhance the focus on short-term fluctuations in the attention matrix. This design improves the length generalization capability of the existing attention-based KT model. The main contributions of this paper can be summarized as follows.

- We propose a decomposition-based layer and kernelized bias-enhanced attention score computation to improve the attention mechanism in current KT models.
- We analyze existing problems with attention-based KT models from a theoretical perspective and explain why the attention score penalty can enhance the length generalization capability of the model. This provides valuable insights for KT-related research.
- We perform extensive and rigorous experiments on three real-world datasets. The results demonstrate that our FlucKT model significantly enhances length generalization and improves prediction performance.

Related Work

Knowledge Tracing

KT has advanced significantly with deep learning and innovative techniques. DKT first introduced RNNs to model student learning (Piech et al. 2015). DKVMN added a dual memory structure for greater accuracy (Zhang et al. 2017). GKT utilized graph neural networks to structure knowledge as a graph (Nakagawa, Iwasawa, and Matsuo 2019). SAKT applied self-attention to address data sparsity and improve prediction accuracy (Pandey and Karypis 2019). SAINT used an encoder-decoder architecture to effectively model exercise-response relationships (Choi et al. 2020). LPKT incorporated learning gains and forgetting effects for better predictions (Shen et al. 2021). Attention-based KT methods, such as ATKT (Guo et al. 2021), SimpleKT (Liu et al. 2022a), and AT-DKT (Liu et al. 2023), have shown strong results, with enhancements like adversarial training, auxiliary tasks, and linear bias mechanisms (Im et al. 2023). DTransformer and extraKT further refined KT by improving temporal dependencies and length generalization (Yin et al. 2023) (Li et al. 2024).

Frequency Domain Analysis

Frequency domain analysis (FDA) refers to the examination of mathematical functions or signals with respect to frequency, rather than time. This analysis is typically used to understand how different frequency components contribute to the overall signal. In practice, signals can be converted from the time domain to the frequency domain using various mathematical transformations, such as the Fourier Transform (Baxes 1994; Peng, Sugiyama, and Mine 2022; Cheung et al. 2020). FDA allows for the analysis and processing of signals-based on their frequency components, which often simplifies complex data and reveals patterns that are not easily detectable in the time domain. FDA

typically considers that data contains both high-frequency and low-frequency signals: high-frequency signals usually represent rapidly changing components in the data, while low-frequency signals represent slowly varying components. In image processing, high-frequency signals correspond to edges, details, and noise, while low-frequency signals correspond to the general contours and colors of the image (Rao et al. 2021; Xu et al. 2020; Suvorov et al. 2022). In time series data, high-frequency components may reflect rapid fluctuations or short-term changes, while low-frequency components reflect long-term trends and cyclic variations (Wu et al. 2021; Zhou et al. 2022).

Preliminaries

In online learning systems, the behavior of a learner is primarily composed of interaction records, which include a sequence of questions and the corresponding responses. For a given learner at the time step t, he/she will answer a question $q_t \in \mathbb{Q}$ drawn from a knowledge components (KCs) $c_t \in \mathbb{C}$, and receive a response $r_t \in \{0, 1\}$. Here, $r_t = 1$ indicates the learner answered the question correctly, while $r_t = 0$ indicates an incorrect answer. Thus, for each learner, we have their interaction records as a sequence:

$$\{(q_1, c_1, r_1), ..., (q_T, c_T, r_T)\}, q_t \in \mathbb{Q}, c_t \in \mathbb{C}, r_t \in \{0, 1\}, \tag{1}$$

where T is the length of the learning sequence, $\mathbb{Q}$ is the set of all questions, $\mathbb{C}$ is the set of all knowledge concepts.

Knowledge Tracing Task: Given the previous interaction records of a learner before time step t as a sequence $\{(q_1, c_1, r_1), ..., (q_T, c_T, r_T)\}$, the objectives of knowledge tracing are: 1) to trace the internal knowledge state z_t of the learner at time step t; 2) to predict their response $\hat{r}_{t+1}$ to the next question q_{t+1}.

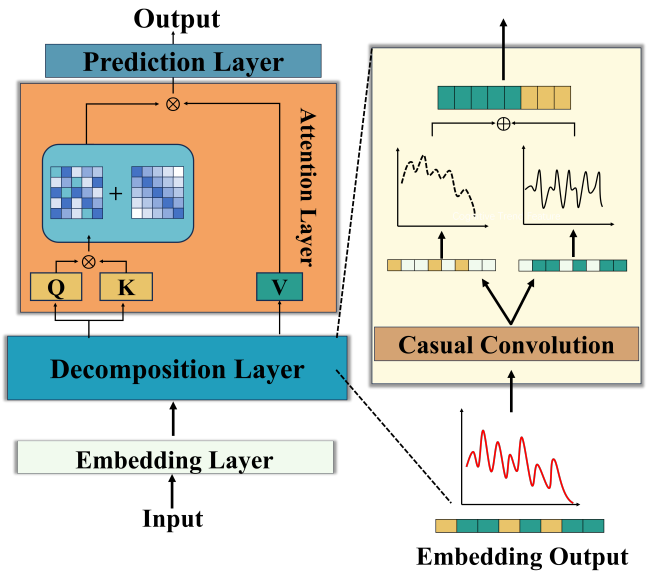


Figure 3: The overview of the proposed FlucKT framework.

Proposed Method

In this section, we introduce FlucKT, a short-term cognitive fluctuation-enhanced attention network for knowledge tracing. As shown in Fig. 3, FlucKT encodes learners' sequential interactions through an embedding layer. It incorporates short-term cognitive fluctuations by using causal convolution to decompose interaction embeddings into long-term trends and short-term fluctuations, which are adaptively aggregated. The attention layer further applies a kernelized bias to penalize long-term attention scores, focusing the attention distribution and improving length generalization during testing.

Embedding Layer

Given that questions associated with the same KCs exhibit varying difficulty levels, it is essential to effectively represent student interactions. In line with AKT (Ghosh, Heffernan, and Lan 2020), SimpleKT (Liu et al. 2022a), DTransformer (Yin et al. 2023), we represent the interaction sequences $\{(q_1, c_1, r_1), ..., (q_T, c_T, r_T)\}$ as follows:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{d}_{q_t} \odot \mathbf{v}_{c_t} \oplus \mathbf{e}_{c_t}, \\ \mathbf{y}_t &= \mathbf{d}_{q_t} \odot \mathbf{v}_{(c_t, r_t)} \oplus \mathbf{e}_{(c_t, r_t)} \end{aligned} \tag{2}$$

where, $\mathbf{x}_t$ denote the latent representations of question q_t and its related KC c_t at the timestamp t. $\mathbf{d}_{q_t}$ represents a learnable question difficulty. $\mathbf{v}_{c_t}$ is the KC variation and $\mathbf{e}_{c_t}$ is the n-dimensional one-hot embeddings of c_t. The symbols $\odot$ and $\oplus$ denote the element-wise multiplication and addition operations, respectively. $\mathbf{y}_t$ represents the augmented representation of $\mathbf{x}_t$ by considering response r_t to the question q_t. $\mathbf{e}_{(c_t, r_t)}$ denotes the embeddings of c_t and r_t. $\mathbf{v}_{(c_t, r_t)}$ represents the KC-response variation of q_t covering this KC c_t with response r_t.

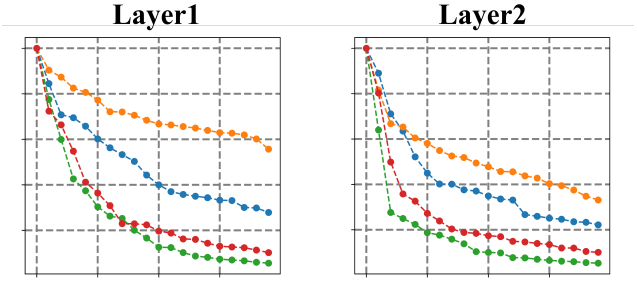


Figure 4: Visualization of the changes in the top 20 largest eigenvalues in the attention matrix as the number of layers increases (1⇒2) in the attention-based KT model. The x-axis represents the index, while the y-axis denotes the normalized magnitude. The blue dotted line corresponds to SimpleKT, the orange dotted line to AKT, the green dotted line to SAINT, and the red dotted line to SAKT.

Fluctuations Enhanced Attention Encoder

After efficiently representing the questions, KCs, and responses in KT, the next step is to capture the students' knowledge or cognitive state. Existing attention-based KT

models first use scaled dot-product operations to calculate correlations between input question sequences ($\mathbf{x}_t$) as attention scores. These attention scores are then weighted and summed with the students' response $\mathbf{y}_t$ to derive the distribution of students' knowledge states across different questions:

$$\mathbf{H} = SoftMax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \tag{3}$$

where $\mathbf{Q} = \mathbf{x}_t\mathbf{W}_Q$, $\mathbf{K} = \mathbf{x}_t\mathbf{W}_K$, and $\mathbf{V} = \mathbf{y}_t\mathbf{W}_V$, and d is the scale factor. For the l-th attention layer, denoting its output of interaction sequence as $\mathbf{H}^{(l)}$, we generalize attention operation in KT as follows:

$$\mathbf{H}^{(l)} = \mathbf{A}\mathbf{H}^{(l-1)}\mathbf{W}_V^{(l)}, \tag{4}$$

where $\mathbf{H}^{(0)} = \mathbf{z}$, $\mathbf{A} = SoftMax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})$.

In this context, $\mathbf{H}$ represents the cognitive state features of a student, which inherently include both long-term cognitive trends and short-term cognitive fluctuations. These two kinds of features are intertwined and entangled. We denote the long-term cognitive trend features as $\mathbf{h}_l$ and the short-term cognitive fluctuation features as $\mathbf{h}_s$, which satisfies that $\mathbf{H} = \mathbf{h}_s + \mathbf{h}_l$ and $\mathbf{h}_s \neq \mathbf{h}_l$. After one layer of attention, $\mathbf{h}_{l/s}$ is represented as $\mathbf{A}\mathbf{h}_{l/s}$ (Here we use $\mathbf{h}_{l/s}$ represents $\mathbf{h}_l$ or $\mathbf{h}_s$). After l layers attention, it is represented as $\mathbf{A}^l\mathbf{h}_{l/s}$. The cosine similarity between $\mathbf{A}^l\mathbf{h}_{l/s}$ and $\mathbf{h}_{l/s}$, denoted as $\cos\left(\langle \mathbf{A}^l\mathbf{h}_{l/s}, \mathbf{h}_{l/s}\rangle\right)$, reflects the similarity between original cognitive features and the cognitive features after l layers of attention.

Theorem 1 Let $\mathbf{A}$ be the self-attention score matrix. For long-term cognitive trend feature $\mathbf{h}_l$ and short-term cognitive fluctuation feature $\mathbf{h}_s$, we have:

$$\lim_{l\to\infty} \cos\left(\langle \mathbf{A}^l\mathbf{h}_s, \mathbf{A}^l\mathbf{h}_l\rangle\right) = 1 \tag{5}$$

Theorem 1 shows that a deeper attention architecture causes the long-term cognitive trend features and short-term cognitive fluctuation features in the cognitive state to become indistinguishable (see Appendix A1 for the proof of Theorem 1)[1]. As shown in Fig. 4, we can observe that as the number of layers increases, the singular values in the attention score matrix tend to decay rapidly (Luo et al. 2024). According to Lemma 1 in Appendix A1, this leads to the correlations between the question sequences $\mathbf{x}_t$ being dominated by the eigenvectors corresponding to the larger singular values. This means that the attention score matrix will ultimately be determined by the frequently occurring patterns in the question sequences, causing the output cognitive state features to lean towards the long-term cognitive trend.

Cognitive Feature Decomposition Layer To address the problem of oversmoothing cognitive features in KT caused by existing attention mechanisms, we draw inspiration from FDA and explicitly decompose the cognitive features represented by the embedding layer. To avoid global information leakage, we use the Wavelet Transform (Oord et al. 2016) to decompose the cognitive features. Specifically, we first apply convolution to $\mathbf{x}_t$ and $\mathbf{y}_t$ to filter out their long-term cognitive trend features. Then, we subtract the long-term cognitive trend features from the original input to obtain the separated short-term cognitive fluctuation features. Suppose that $\mathbf{x}_t, \mathbf{y}_t \in \mathbb{R}^{B\times L\times D}$, we have:

$$\begin{aligned} \mathbf{x}_t^{'} &= P(\mathbf{x}_t, [0,2,1]), \mathcal{L}[\mathbf{x}_t] = C(\mathbf{x}_t^{'}), \\ \mathcal{L}[\mathbf{x}_t] &= P(\mathcal{L}[\mathbf{x}_t], [0,2,1]), \mathcal{S}[\mathbf{x}_t] = \mathbf{x}_t - \mathcal{L}[\mathbf{x}_t], \end{aligned} \tag{6}$$

where P and C denote operations of permute, and casual convolution operations, respectively. B, L, D represent the batch size, input sequence length, and embedding size. $\mathcal{L}[\cdot]$ and $\mathcal{S}[\cdot]$ represent long-term cognitive trend features and short-term cognitive fluctuation features. We only present the decomposition of $\mathbf{x}_t$; the decomposition of $\mathbf{y}_t$ follows the same process as that of $\mathbf{x}_t$. Finally, we aggregate the long-term cognitive trend features and short-term cognitive fluctuation features adaptively as follows:

$$\mathbf{x}_t = \mathcal{L}[\mathbf{x}_t] + \boldsymbol{\mu}\mathcal{S}[\mathbf{x}_t], \mathbf{y}_t \quad = \mathcal{L}[\mathbf{y}_t] + \boldsymbol{\nu}\mathcal{S}[\mathbf{y}_t], \tag{7}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ represent the learnable aggregated parameters. Through this explicit decomposition operation, we forcibly separate the two entangled cognitive features and then reassemble them with weighted proportions, enhancing the representation of short-term cognitive fluctuation features in the original input. Additionally, by using learnable weights, the attention mechanism can adjust its focus on the two types of cognitive features according to the downstream tasks.

Kernelized Bias Enhanced Attention Scores In practical applications of KT, we aim for a model that can be trained on limited student interaction data and still make stable predictions on longer sequences (e.g., as users' answering data updates) without requiring fine-tuning. However, the suppression of short-term cognitive fluctuation features by the attention mechanism also affects its length generalization capability: excessive focus on long-term cognitive trend features prevents the model from effectively predicting short-term cognitive fluctuation features in unknown sequences. To effectively enhance the length generalization ability of attention-based KT models, a reasonable approach is to penalize the attention values to make them more attentive to local information (corresponding to high-frequency signals). Inspired by previous studies (Press, Smith, and Lewis 2021; Chi et al. 2022), we utilize kernelized bias to penalize the attention scores, thereby improving its modeling capability for local information. Simultaneously, we have also analyzed from the perspective of entropy invariance (Su 2022) why this penalization approach on attention enhances its length generalization capability and why kernelized bias is superior to linear bias (See Appendix A2). Specifically, we model the positional differences between tokens in attention using conditionally positive definite kernels. In Eq.(3), if $a_{i,j}$ is an element of $\mathbf{A}$, we have

$$a_{i,j} = \frac{\exp(\epsilon q_i \cdot k_j)}{\sum_{j=1}^{n}\exp(\epsilon q_i \cdot k_j)}, \tag{8}$$

[1]The full text of FlucKT can be found at https://pykt.org/.

Dataset	#Students	#Concepts	#Questions	#Interactions	Avg. interactions per student	Percentage of length ≥ 200
AL2005	574	112	173,113	607,021	1,057.5	81.71%
BD2006	1,145	493	129,263	1,817,458	1,587.3	92.75%
NIPS34	4,918	57	948	1,382,678	281.1	58.72%

Table 1: Statistics of the datasets.

where $\epsilon = \sqrt{D}$, where D is the dimension of the query/key vectors. Our modified attention scores can be defined as:

$$a_{i,j} = \frac{\exp(\epsilon q_i k_j - \tau_1 \log(1 + \tau_2 |i-j|))}{\sum_{j=1}^{n} \exp(\epsilon q_i k_j - \tau_1 \log(1 + \tau_2 |i-j|))}, \quad (9)$$

where τ_1 and τ_2 are two learnable parameters that satisfy $0 < \tau_1 \leq 1$, and $0 < \tau_2 \leq 2$. $exp(\cdot)$ equals $e^{(\cdot)}$.

Prediction Layer

After L attention layers that hierarchically extract knowledge state information from previous interactions, we obtain the final combined representation of behavior sequences. Denoting the learned representations as $\mathbf{h}_{t+1}$, we construct a two-layer fully connected neural network as the prediction layer to forecast student responses. To optimize the predictive function, we minimize the binary cross-entropy loss between the actual student response $\mathbf{r}_{t+1}$ and the predicted response $\hat{\mathbf{r}}_{t+1}$ (Liu et al. 2022b,a). This prediction layer ensures that our model effectively learns to estimate the probability of a student answering correctly, thereby enhancing its predictive performance, which is defined as follows:

$$\begin{aligned} \hat{\mathbf{r}}_{t+1} &= \gamma(\delta(\mathbf{W}_2 \cdot \delta(\mathbf{W}_1 \cdot [\mathbf{h}_{t+1}; \mathbf{x}_{t+1}] + \mathbf{b}_1) + \mathbf{b}_2)), \\ Loss &= -\sum_t (\mathbf{r}_{t+1} \cdot \log \hat{\mathbf{r}}_{t+1} + (1 - \mathbf{r}_{t+1}) \cdot \log(1 - \hat{\mathbf{r}}_{t+1})), \end{aligned} \quad (10)$$

where γ and δ denote Sigmoid and Relu functions. $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{W}_1$, $\mathbf{W}_2$ are trainable parameters.

Experiments

In this section, we first introduce the experimental setup, including the three real-world datasets used, the baseline methods, and the implementation details. We then analyze the experimental results to demonstrate the effectiveness of the proposed FlucKT.

Experiential Settings

Datasets We evaluate the effectiveness of FlucKT across three diverse real-world datasets, each representing different learning scenarios. Table 1 presents the statistics for all datasets. More detailed information of these three datasets can be found at Appendix A3.1.

Baselines To demonstrate that our proposed FlucKT framework effectively enhances the robustness of current knowledge tracing methods, we selected 13 state-of-the-art knowledge tracing methods as baselines for comparison, including DKT (Piech et al. 2015), DKVMN (Zhang et al. 2017), GKT (Nakagawa, Iwasawa, and Matsuo 2019), SAKT (Pandey and Karypis 2019), SAINT (Choi et al. 2020), AKT (Ghosh, Heffernan, and Lan 2020), ATKT (Guo et al. 2021), LPKT (Shen et al. 2021), SimpleKT (Liu et al. 2022a), AT-DKT (Liu et al. 2023), FoLiBiKT (Im et al. 2023), DTransformer (Yin et al. 2023), and extraKT (Li et al. 2024).

Experiential Results

Model	AUC				
	Length of Interaction Sequences				
	200	400	600	800	1000
DKT	0.8149	0.8150	0.8150	0.8149	0.8149
DKVMN	0.8054	0.8039	0.8030	0.8025	0.8023
GKT	0.8110	0.8111	0.8111	0.8111	0.8111
SAKT	0.7899	0.6743	0.6691	0.6677	0.6666
SAINT	0.7715	0.6691	0.6589	0.6539	0.6551
AKT	0.8306	0.8277	0.8258	0.8241	0.8227
ATKT	0.7995	0.7816	0.7641	0.7523	0.7446
LPKT	0.8268	0.8216	0.8107	0.7990	0.7891
SimpleKT	0.8210	0.7808	0.7763	0.7535	0.7655
AT-DKT	0.8246	0.8238	0.8235	0.8233	0.8233
FoLiBiKT	0.8310	0.8288	0.8272	0.8256	0.8242
DTransformer	0.8188	0.8156	0.8137	0.8123	0.8112
extraKT	0.8317	0.8317	0.8317	0.8317	0.8317
FlucKT	**0.8376**	**0.8370**	**0.8365**	**0.8360**	**0.8358**

Table 2: AUC results on AL2005 dataset.

Model	AUC				
	Length of Interaction Sequences				
	200	400	600	800	1000
DKT	0.8015	0.8015	0.8015	0.8015	0.8015
DKVMN	0.7983	0.7956	0.7936	0.7925	0.7919
GKT	0.8046	0.8047	0.8047	0.8047	0.8047
SAKT	0.7739	0.7097	0.7000	0.6987	0.6962
SAINT	0.7791	0.6847	0.6816	0.6692	0.6697
AKT	0.8208	0.8187	0.8168	0.8155	0.8144
ATKT	0.7889	0.7641	0.7370	0.7142	0.6963
LPKT	0.8056	0.8014	0.7965	0.7939	0.7923
SimpleKT	0.8151	0.7897	0.7764	0.7726	0.7724
AT-DKT	0.8104	0.8098	0.8095	0.8092	0.8089
FoLiBiKT	0.8199	0.8171	0.8145	0.8125	0.8110
DTransformer	0.8093	0.8052	0.8023	0.8002	0.7985
extraKT	0.8247	0.8246	0.8246	0.8245	0.8245
FlucKT	**0.8269**	**0.8263**	**0.8257**	**0.8253**	**0.8250**

Table 3: AUC results on BD2006 dataset.

Overall Performance Tables 2-4 present the overall performance of various models, including our proposed FlucKT model, on the AL2005, BD2006, and NIPS34 datasets in terms of AUC (The complete AUC and ACC results can be found in Appendix A3.2.). The best AUC and ACC values are highlighted in bold, while the second-best are under-

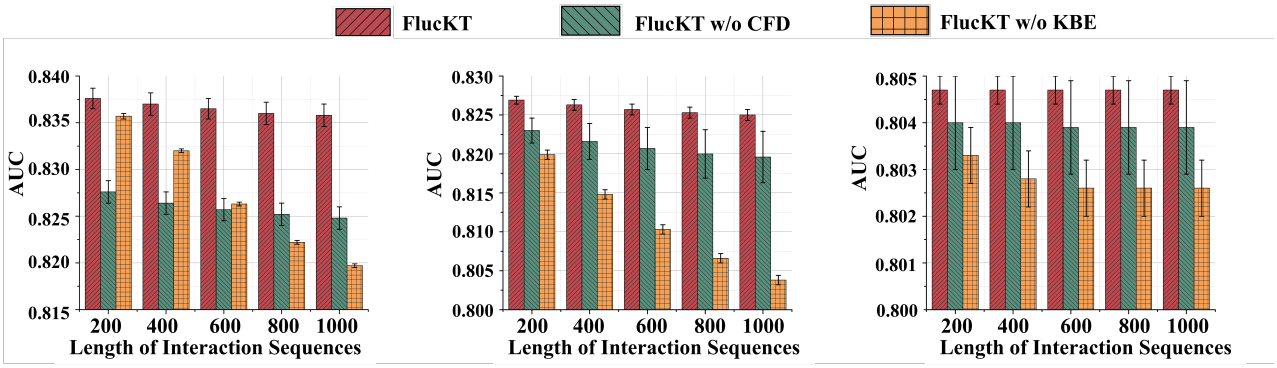


Figure 5: Ablation study results (FlucKT) in terms of AUC on three datasets.

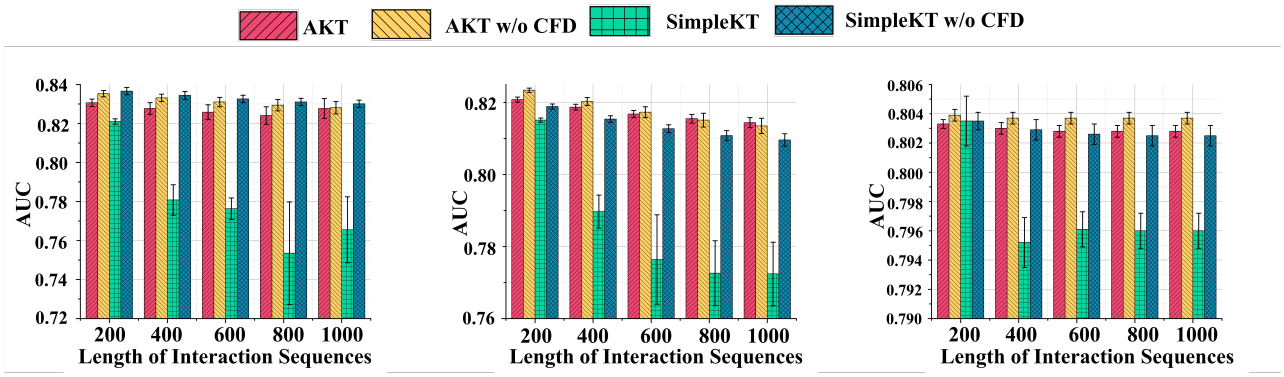


Figure 6: Ablation study results (AKT and SimpleKT) in terms of AUC on three datasets.

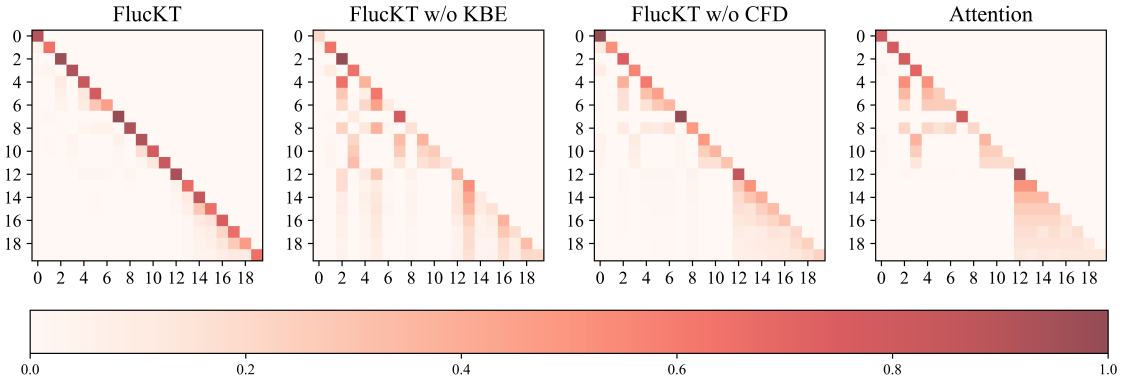
Model	AUC				
	Length of Interaction Sequences				
	200	400	600	800	1000
DKT	0.7689	0.7689	0.7689	0.7689	0.7689
DKVMN	0.7673	0.7673	0.7673	0.7672	0.7672
GKT	0.7689	0.7689	0.7689	0.7689	0.7689
SAKT	0.7525	0.7331	0.7329	0.7330	0.7330
SAINT	0.7895	0.7708	0.7703	0.7700	0.7700
AKT	0.8033	0.8030	0.8028	0.8028	0.8028
ATKT	0.7665	0.7630	0.7620	0.7619	0.7619
LPKT	0.8004	0.7997	0.7993	0.7992	0.7992
SimpleKT	0.8035	0.7952	0.7961	0.7960	0.7960
AT-DKT	0.7816	0.7815	0.7815	0.7815	0.7815
FoLiBiKT	0.8032	0.8029	0.8028	0.8028	0.8028
DTransformer	0.7994	0.7988	0.7985	0.7985	0.7985
extraKT	0.8045	0.8047	0.8047	0.8047	0.8047
FlucKT	**0.8047**	**0.8047**	**0.8047**	**0.8047**	**0.8047**

Table 4: AUC results on NIPS34 dataset.

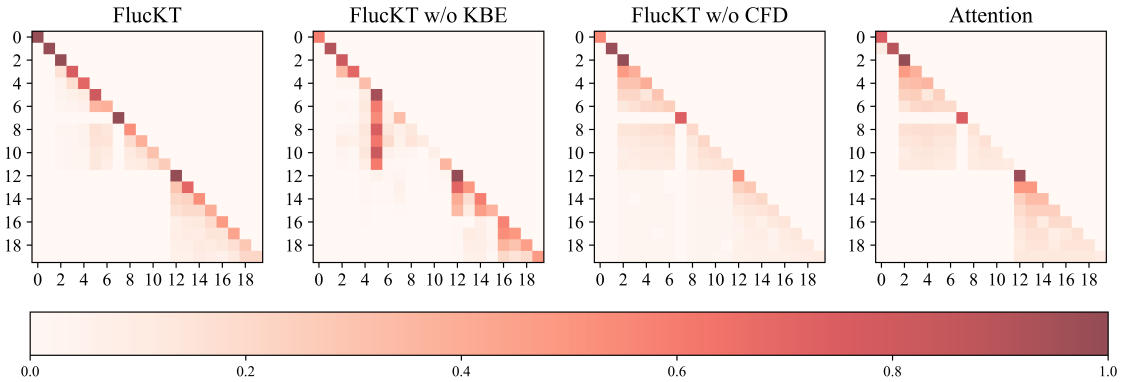
lined. From these tables, we observe that: 1) In the AL2005 dataset, the FlucKT model achieves the highest AUC and ACC values across almost all context window sizes. AUC values range from 0.8376 at a window size of 200 to 0.8358 at a window size of 1000. ACC values range from 0.8153 at a window size of 200 to 0.8147 at a window size of 1000, indicating robust performance and effective knowledge extraction. 2) For the BD2006 dataset, the FlucKT model consistently shows superior performance. AUC values range from 0.8269 at a window size of 200 to 0.8252 at a window size of 1000. ACC values range from 0.8614 at a window size of 200 to 0.8609 at a window size of 1000, demonstrating stability and reliability in its predictions across different window sizes. 3) On the NIPS34 dataset, the FlucKT model achieves the best AUC scores across all context window sizes, with values consistently around 0.8047. Although the ACC values are competitive, they are slightly lower than the top performing extraKT model by only 0.0003, maintaining a consistent score of 0.7337 for all window sizes. The average number of student interactions in the NIPS34 dataset is the lowest among the three datasets (see Table 1), which explains why FlucKT's performance on NIPS34 is less prominent. However, the results still demonstrate FlucKT's robustness in handling datasets with fewer student interactions.

While the AUC improvement in Tables 2-4 is less than 1%, it remains significant at a context window size of 200. Notably, recent benchmarks report only a 3.5% KT prediction gain since 2015, with many results deemed unreliable due to flawed evaluations. Our study strictly adheres to pyKT (Liu et al. 2022b) evaluation protocols and includes comprehensive hyperparameter tuning for all baselines.

Ablation Study We analyze the effect of two key components in FlucKT via ablation studies on three datasets: CFD (Cognitive Feature Decomposition layer) and KBE (Kernelized Bias Enhanced attention). Results are reported in terms of AUC (Fig. 5-6, and Appendix A3.2 for ACC). Key findings include: 1) Removing either CFD or KBE decreases FlucKT's performance, indicating both are essential. 2) CFD



(a) Averaged attention scores on the 1st layer.



(b) Averaged attention scores on the 2nd layer.

Figure 7: Visualization of cognitive feature decomposition (CFD) and kernelized bias enhanced attention scores (KBE) impact on attention scores.

has a stronger impact on AL2005, while KBE influences the other two datasets more significantly. 3) CFD also benefits other attention-based KT models. For example, integrating CFD into AKT and SimpleKT improves their AUC (Fig. 6) and ACC (Fig. 9), particularly enhancing SimpleKT's length generalization ability over 200-1000 steps. This demonstrates CFD's utility in improving attention-based KT models and their generalization capabilities.

Visualization To qualitatively analyze the impact of the two improvements in FlucKT (namely, CFD and KBE), we visualized the attention scores, as shown in Fig. 7. We observed the following: 1) FlucKT causes the distribution of attention scores to become more concentrated, indicating that FlucKT pays greater attention to short-term fluctuation features. However, in the second layer, the attention scores of FlucKT also expand to some extent, demonstrating that FlucKT integrates cognitive trend features with cognitive fluctuation features; 2) It is observable that KBE (FlucKT w/o CFD) indeed penalizes attention-based on distance, thereby making the attention more focused; 3) By comparing FlucKT w/o KBE' and 'Attention' in Fig. 7 (a) and (b), we can see that the design of CFD (FlucKT w/o KBE) reorganizes the distribution of attention scores to some extent: certain long-distance attention scores are strengthened, while some short-distance attention scores are weakened. This aligns with the core design of CFD, which allows the KT model to adaptively determine the fusion weights for cognitive trend features and cognitive fluctuation features-based on the performance of downstream tasks. Additionally, we conducted a case study to qualitatively compare the performance of FlucKT and its variants (See Appendix A3.2).

Conclusion

This paper presents FlucKT, an enhanced attention-based KT model designed to address the shortcomings of existing models in capturing short-term cognitive fluctuations. By introducing a decomposition-based layer and a kernelized bias attention score mechanism, FlucKT improves both prediction accuracy and length generalization across various datasets. Our findings demonstrate the importance of accounting for both long-term trends and short-term fluctuations in KT tasks. FlucKT effectively balances these cognitive features, resulting in a more robust and accurate model. These advances contribute to the development of more effective personalized education systems.

Acknowledgments

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 2023 Industry-University-Research Innovation Fund for Chinese Universities (2023KY005).

References

Barabási, A.-L.; Albert, R.; and Jeong, H. 1999. Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications*, 272(1–2): 173–187.

Baxes, G. A. 1994. *Digital image processing: Principles and applications*. John Wiley & Sons, Inc.

Cheung, M.; Shi, J.; Wright, O.; Jiang, L. Y.; Liu, X.; and Moura, J. M. 2020. Graph signal processing and deep learning: Convolution, pooling, and topology. *IEEE Signal Processing Magazine*, 37(6): 139–149.

Chi, T.-C.; Fan, T.-H.; Ramadge, P. J.; and Rudnicky, A. 2022. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35: 8386–8399.

Choi, Y.; Lee, Y.; Cho, J.; Baek, J.; Kim, B.; Cha, Y.; Shin, D.; Bae, C.; and Heo, J. 2020. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the 7th ACM Conference on Learning at Scale*, 341–344.

Ghosh, A.; Heffernan, N.; and Lan, A. S. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2330–2339.

Guo, X.; Huang, Z.; Gao, J.; Shang, M.; Shu, M.; and Sun, J. 2021. Enhancing knowledge tracing via adversarial training. In *Proceedings of the 29th ACM International Conference on Multimedia*, 367–375.

Im, Y.; Choi, E.; Kook, H.; and Lee, J. 2023. Forgetting-aware linear bias for attentive knowledge tracing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3958–3962.

Li, X.; Bai, Y.; Zheng, Y.; Hou, M.; Zhan, B.; Huang, Y.; Liu, Z.; Gao, B.; and Luo, W. 2024. Extending context window of attention-based knowledge tracing models via length extrapolation. In *Proceedings of the 27th European Conference on Artificial Intelligence*.

Liu, Z.; Liu, Q.; Chen, J.; Huang, S.; Gao, B.; Luo, W.; and Weng, J. 2023. Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the 32nd International World Wide Web Conference*, 4178–4187.

Liu, Z.; Liu, Q.; Chen, J.; Huang, S.; and Luo, W. 2022a. simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *Proceedings of the 11th International Conference on Learning Representations*.

Liu, Z.; Liu, Q.; Chen, J.; Huang, S.; Tang, J.; and Luo, W. 2022b. pyKT: A python library to benchmark deep learning-based knowledge tracing models. *Advances in Neural Information Processing Systems*, 35: 18542–18555.

Luo, R.; Huang, H.; Yu, S.; Zhang, X.; and Xia, F. 2024. FairGT: A fairness-aware graph transformer. *arXiv preprint arXiv:2404.17169*.

Mozer, M. C.; Pashler, H.; Cepeda, N.; Lindsey, R.; and Vul, E. 2009. Predicting the optimal spacing of study: a multi-scale context model of memory. 1321–1329.

Nakagawa, H.; Iwasawa, Y.; and Matsuo, Y. 2019. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 156–163.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Pandey, S.; and Karypis, G. 2019. A self-attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, 384–389.

Peng, S.; Sugiyama, K.; and Mine, T. 2022. Less is more: Reweighting important spectral graph features for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1273–1282.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 28.

Press, O.; Smith, N.; and Lewis, M. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. In *Proceedings of the 9th International Conference on Learning Representations*.

Qin, Z.; Sun, W.; Deng, H.; Li, D.; Wei, Y.; Lv, B.; Yan, J.; Kong, L.; and Zhong, Y. 2022. cosFormer: Rethinking softmax in attention. In *Proceedings of the 9th International Conference on Learning Representations*.

Rao, Y.; Zhao, W.; Zhu, Z.; Lu, J.; and Zhou, J. 2021. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34: 980–993.

Salthouse, T. A.; Nesselroade, J. R.; and Berish, D. E. 2006. 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.

Shen, S.; Liu, Q.; Chen, E.; Huang, Z.; Huang, W.; Yin, Y.; Su, Y.; and Wang, S. 2021. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1452–1460.

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

Su, J. 2022. A quick derivation of softmax entropy invariance. Accessed: 2024-12-11.

Suvorov, R.; Logacheva, E.; Mashikhin, A.; Remizova, A.; Ashukha, A.; Silvestrov, A.; Kong, N.; Goka, H.; Park, K.; and Lempitsky, V. 2022. Resolution-robust large mask inpainting with Fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2149–2159.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

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

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Xu, K.; Qin, M.; Sun, F.; Wang, Y.; Chen, Y.-K.; and Ren, F. 2020. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1740–1749.

Yin, Y.; Dai, L.; Huang, Z.; Shen, S.; Wang, F.; Liu, Q.; Chen, E.; and Li, X. 2023. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the 32nd International World Wide Web Conference*, 855–864.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized Bayesian knowledge tracing models. In *Proceedings of the 16th International Conference on Artificial Intelligence in Education*, 171–180.

Zhang, J.; Shi, X.; King, I.; and Yeung, D.-Y. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International World Wide Web Conference*, 765–774.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, 27268–27286.

Appendix

A1: Proof of Theorem 1

Theorem 1 Let $\mathbf{A}$ be the self-attention score matrix. For long-term cognitive trend feature $\mathbf{h}_l$ and short-term cognitive fluctuation feature $\mathbf{h}_s$, we have:

$$\lim_{l\to\infty} \cos\left(\langle \mathbf{A}^l\mathbf{h}_s, \mathbf{A}^l\mathbf{h}_l\rangle\right) = 1 \tag{11}$$

To prove Theorem 1, we first establish the following lemmas.

Lamma 1. Assume $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a symmetric matrix with real-valued entries. The eigenvalue are ordered as $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_n|$, and $\mathbf{p}_i$ ($i \in \{1, 2, ..., n\}$) are corresponding eigenvectors. Then, the following equation holds:

$$\begin{aligned}
\cos\left(\langle \mathbf{h}, \mathbf{p}_i\rangle\right) &= \frac{\mathbf{h}^\top\mathbf{p}_i}{\sqrt{\sum_{j=1}^n (\mathbf{h}^\top\mathbf{p}_j)^2}} = \frac{\beta_i}{\sqrt{\sum_{j=1}^n \beta_j^2}}, \\
\cos\left(\langle \mathbf{A}\mathbf{h}, \mathbf{p}_i\rangle\right) &= \frac{\beta_i\lambda_i}{\sqrt{\sum_{j=1}^n \beta_j^2\lambda_j^2}},
\end{aligned} \tag{12}$$

where $\beta_i = \mathbf{h}^\top\mathbf{p}_i$ is the weight of $\mathbf{h}$ on $\mathbf{p}_i$.

Proof: Since $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a symmetric matrix, assume the eigendecomposition $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$ with $\mathbf{p}_i$ ($i \in \{1, 2, ..., n\}$) are corresponding eigenvectors. We have:

$$\begin{aligned}
\cos\left(\langle \mathbf{h}, \mathbf{p}_i\rangle\right) &= \frac{\mathbf{h}^\top\mathbf{p}_i}{\|\mathbf{h}\|\|\mathbf{p}_i\|} = \frac{\mathbf{h}^\top\mathbf{p}_i}{\|\mathbf{h}\|}, \\
&= \frac{\mathbf{h}^\top\mathbf{p}_i}{\sqrt{\mathbf{h}^\top\mathbf{h}}} = \frac{\mathbf{h}^\top\mathbf{p}_i}{\sqrt{(\mathbf{P}^\top\mathbf{h})^\top\mathbf{P}^\top\mathbf{h}}}, \\
&= \frac{\mathbf{h}^\top\mathbf{p}_i}{\sqrt{\sum_{j=1}^n (\mathbf{p}_j^\top\mathbf{h})^2}} = \frac{\mathbf{h}^\top\mathbf{p}_i}{\sqrt{\sum_{j=1}^n (\mathbf{h}^\top\mathbf{p}_j)^2}}, \\
&= \frac{\beta_i}{\sqrt{\sum_{j=1}^n \beta_j^2}}
\end{aligned} \tag{13}$$

Moreover, we can prove that:

$$\begin{aligned}
\cos\left(\langle \mathbf{A}\mathbf{h}, \mathbf{p}_i\rangle\right) &= \frac{(\mathbf{A}\mathbf{h})^\top\mathbf{p}_i}{\|\mathbf{A}\mathbf{h}\|\|\mathbf{p}_i\|} = \frac{(\mathbf{A}\mathbf{h})^\top\mathbf{p}_i}{\sqrt{(\mathbf{A}\mathbf{h})^\top\mathbf{A}\mathbf{h}}}, \\
&= \frac{(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))^\top\mathbf{p}_i}{\sqrt{(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))^\top(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))}} = \frac{(\mathbf{P}^\top\mathbf{h})^\top\mathbf{\Lambda}\mathbf{P}^\top\mathbf{p}_i}{\sqrt{(\mathbf{P}^\top\mathbf{h})^\top\mathbf{\Lambda}^2(\mathbf{P}^\top\mathbf{h})}}, \\
&= \frac{\left(\mathbf{p}_1^\top\mathbf{h}, \ldots, \mathbf{p}_i^\top\mathbf{h}, \ldots, \mathbf{p}_n^\top\mathbf{h}\right)\begin{pmatrix}\lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n\end{pmatrix}\begin{pmatrix}\mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_i^\top \\ \vdots \\ \mathbf{p}_n^\top\end{pmatrix}\mathbf{p}_i}{\sqrt{\left(\mathbf{P}^\top\mathbf{h}\right)^\top\lambda^2\left(\mathbf{P}^\top\mathbf{h}\right)}}, \\
&= \frac{\mathbf{p}_i^\top\mathbf{h}\lambda_i}{\sqrt{\sum_{j=1}^n (\mathbf{p}_j^\top\mathbf{h})^2\lambda_j^2}} = \frac{\mathbf{h}^\top\mathbf{p}_i\lambda_i}{\sqrt{\sum_{j=1}^n (\mathbf{h}^\top\mathbf{p}_j)^2\lambda_j^2}} = \frac{\beta_i\lambda_i}{\sqrt{\sum_{j=1}^n \beta_j^2\lambda_j^2}}
\end{aligned} \tag{14}$$

Eq. (12) shows that when the attention mechanism encounters dissimilar eigenvalues, the resulting signals exhibit higher cosine similarity with the eigenvectors associated with larger eigenvalues and lower cosine similarity (orthogonality) with those linked to smaller eigenvalues. This implies that the attention mechanism will ultimately be dominated by the eigenvectors corresponding to the larger eigenvalues. In the context of KT, this leads to the relationships between students' knowledge state sequences being dominated by frequently occurring patterns, namely the cognitive trends. Moreover, this tendency is further amplified in deeper architectures:

Lamma 2. Assume $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a symmetric matrix with real-valued entries. The eigenvalue are ordered as $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_n|$, and $\mathbf{p}_i$ ($i \in \{1, 2, ..., n\}$) are corresponding eigenvectors. Then, for any given $\mathbf{h}, \mathbf{h}'$, we have:

$$\begin{aligned}
&|\cos\left(\langle \mathbf{A}^{l+1}, \mathbf{p}_1\rangle\right)| \geq |\cos\left(\langle \mathbf{A}^l, \mathbf{p}_1\rangle\right)| \; and \\
&\|\cos\left(\langle \mathbf{A}^{l+1}, \mathbf{p}_n\rangle\right)| \leq |\cos\left(\langle \mathbf{A}^l, \mathbf{p}_n\rangle\right)| \; for \\
&l = 0, 1, 2, ..., +\infty, \\
&if\ |\lambda_1| > |\lambda_2|, \lim_{l\to\infty} \cos\left(\langle \mathbf{A}^l\mathbf{h}, \mathbf{A}^l\mathbf{h}'\rangle\right) = \\
&\lim_{l\to\infty} |\cos\left(\langle \mathbf{A}^{l+1}\mathbf{h}, \mathbf{p}_1\rangle\right)|.
\end{aligned} \tag{15}$$

Proof: As $\mathbf{A}^l = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$ and Lamma 1, for $l = 0, 1, 2, ... + \infty$, we have:

$$\begin{aligned}
|\cos\left(\langle \mathbf{A}^l\mathbf{h}, \mathbf{p}_1\rangle\right)| &= \frac{|\beta_1\lambda_1^k|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2k}}} \\
&= \frac{|\beta_1|}{|\lambda_1|}\frac{|\beta_1\lambda_1^k|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2k}}} = \frac{|\beta_1\lambda_1^{k+1}|}{\sqrt{\beta_1^2\sum_{i=1}^n \beta_i^2\lambda_i^{2k}}} \\
&= \frac{|\beta_1\lambda_1^{k+1}|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2(k+1)}}} \leq \frac{|\beta_1\lambda_1^{k+1}|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2(k+1)}}} \\
&= |\cos(\langle \mathbf{A}^{l+1}\mathbf{h}, \mathbf{p}_1\rangle)|.
\end{aligned} \tag{16}$$

Similarly, we can prove that $\|\cos\left(\langle \mathbf{A}^l, \mathbf{p}_n\rangle\right)| \geq |\cos\left(\langle \mathbf{A}^{l+1}, \mathbf{p}_n\rangle\right)|$.

Since $|\cos\left(\langle \mathbf{A}^l\mathbf{h}, \mathbf{p}_n\rangle\right)|$ monotonously increases with respect to k and has the upper bound 1, $|\cos\left(\langle \mathbf{A}^l\mathbf{h}, \mathbf{p}_n\rangle\right)|$ must be convergent. We have:

$$\begin{aligned}
\lim_{l\to\infty} \left|\cos\left(\langle S^l\mathbf{h}, \mathbf{p}_1\rangle\right)\right| &= \lim_{l\to\infty} \frac{|\beta_1\lambda_1^l|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2l}}} \\
&= \lim_{l\to\infty} \frac{|\beta_1|}{\sqrt{\beta_1^2 + \sum_{i=2}^n \beta_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}} \\
&= \frac{|\beta_1|}{\sqrt{\beta_1^2 + \lim_{l\to\infty}\sum_{i=2}^n \lambda_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}}
\end{aligned} \tag{17}$$

As $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n|$, we have $\lim_{l\to\infty}\sum_{i=2}^n \beta^2(\frac{\lambda_i}{\lambda_1})^{2k} = 0$ and the convergence speed is

decided by $|\frac{\lambda_2}{\lambda_1}|$. Therefore, $\lim_{l\to\infty} |\cos(\langle \mathbf{A}^{l+1}\mathbf{h}, \mathbf{p}_1\rangle)| = 1$.

$$
\begin{aligned}
&\cos(\langle \mathbf{A}^l\mathbf{h}, \mathbf{A}^l\mathbf{h}'\rangle) = \frac{(\mathbf{A}\mathbf{h})^\top \mathbf{A}\mathbf{h}'}{\|\mathbf{A}\mathbf{h}\|\|\mathbf{A}\mathbf{h}'\|} \\
&= \frac{(\mathbf{A}\mathbf{h})^\top \mathbf{A}\mathbf{h}'}{\sqrt{(\mathbf{A}\mathbf{h})^\top \mathbf{A}\mathbf{h}}\sqrt{(\mathbf{A}\mathbf{h}')^\top \mathbf{A}\mathbf{h}'}} \\
&= \frac{(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))^\top \mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}')}{\sqrt{(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))^\top (\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}))}\sqrt{(\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}'))^\top (\mathbf{P}\mathbf{\Lambda}(\mathbf{P}^\top\mathbf{h}'))}} \\
&= \frac{(\mathbf{P}^\top\mathbf{h})^\top \mathbf{\Lambda}^2\mathbf{P}^\top\mathbf{h}'}{\sqrt{(\mathbf{P}^\top\mathbf{h})^\top \mathbf{\Lambda}^2(\mathbf{P}^\top\mathbf{h})}\sqrt{(\mathbf{P}^\top\mathbf{h}')^\top \mathbf{\Lambda}^2(\mathbf{P}^\top\mathbf{h}')}} \\
&= \frac{\boldsymbol{\beta}^\top \mathbf{\Lambda}^2\boldsymbol{\gamma}}{\sqrt{\boldsymbol{\beta}^\top \mathbf{\Lambda}^2\boldsymbol{\beta}}\sqrt{\boldsymbol{\gamma}^\top \mathbf{\Lambda}^2\boldsymbol{\gamma}}} = \frac{\sum_{i=1}^n \beta_i\gamma_i\lambda_i^2}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^2}\sqrt{\sum_{i=1}^n \gamma_i^2\lambda_i^2}}
\end{aligned}
\tag{18}
$$

Then,

$$
\begin{aligned}
&\lim_{l\to\infty} \left|\cos\left(\left\langle \mathbf{A}^l\mathbf{h}, \mathbf{A}^l\mathbf{h}'\right\rangle\right)\right| \\
&= \lim_{l\to\infty} \frac{\left|\sum_{i=1}^n \beta_i\gamma_i\lambda_i^{2l}\right|}{\sqrt{\sum_{i=1}^n \beta_i^2\lambda_i^{2l}}\sqrt{\sum_{i=1}^n \gamma_i^2\lambda_i^{2l}}} \\
&= \lim_{l\to\infty} \frac{\left|\sum_{i=1}^n \beta_i\gamma_i \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}\right|}{\sqrt{\sum_{i=1}^n \beta_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}\sqrt{\sum_{i=1}^n \gamma_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}} \\
&= \lim_{l\to\infty} \frac{\left|\beta_1\gamma_1 + \sum_{i=2}^n \beta_i\gamma_i \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}\right|}{\sqrt{\beta_1^2 + \sum_{i=2}^n \beta_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}\sqrt{\gamma_1^2 + \sum_{i=2}^n \gamma_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2l}}} \\
&= \frac{|\beta_1\gamma_1|}{\sqrt{\beta_1^2}\sqrt{\gamma_1^2}} \\
&= 1
\end{aligned}
\tag{19}
$$

Based on Lemma 2, we can derive the conclusion of Theorem 1. In summary, after passing through l layers of attention, the distinctions between the trend features and fluctuation features in KT data are smoothed out. The entire attention matrix becomes determined by the larger eigenvalues and their corresponding eigenvectors.

A2:Analysis of the Length Generation Capability of Attention-Based Models

The scaled dot-product attention can be rewritten as follows:

$$
\mathbf{o_i} = \sum_{j=1}^n a_{i,j}\mathbf{v_j}, \quad a_{i,j} = \frac{e^{\lambda \mathbf{q_i}\cdot\mathbf{k_j}}}{\sum_{j=1}^n e^{\lambda \mathbf{q_i}\cdot\mathbf{k_j}}}, \tag{20}
$$

where $\lambda = \frac{1}{\sqrt{d}}$. One perspective on the length generalization capability of attention-based models presented in this paper is that, to *enhance the generalization of the results to unknown lengths, the design of the attention mechanism should ensure that $a_{i,j}$ possesses entropy invariance*.

Specifically, $a_{i,j}$ can be regarded as a conditional distribution where i is the condition and j is the random variable, with its entropy being:

$$
\mathcal{H} = -\sum_{j=1}^n a_{i,j}\log a_{i,j} \tag{21}
$$

Entropy invariance means that $\mathcal{H}$ should be insensitive to the length n. More specifically, if additional tokens are appended to the existing tokens, the newly calculated $a_{i,j}$ will naturally change, but we hope that $\mathcal{H}$ does not change significantly. We aim for entropy invariance to ensure that after introducing new tokens, the existing tokens can still focus on the original tokens in the same manner. We do not want the introduction of new tokens to excessively 'dilute' the original attention, leading to a significant change in the summation result.

Next, we will demonstrate that introducing attention penalties through a bias approach can better ensure entropy invariance of the attention matrix during length generalization. Here, we define the dimensions of attention score matrix is $n \times n$ and the indices i and j correspond to the row and column positions within the matrix, respectively, where $0 \le i \le n$ and $0 \le j \le n$.

First, assume $\mathbf{q}_i\mathbf{k}_j = \mathbf{s}_i$, we have:

$$
p_i = \frac{e^{\lambda \mathbf{s}_i}}{\sum_{i=1}^n e^{\lambda \mathbf{s}_i}} \tag{22}
$$

The entropy is:

$$
\begin{aligned}
\mathcal{H} &= -\sum_{i=1}^n p_i \log p_i \\
&= \log\sum_{i=1}^n e^{\lambda \mathbf{s}_i} - \lambda\sum_{i=1}^n p_i\mathbf{s}_i \\
&= \log n + \log\frac{1}{n}\sum_{i=1}^n e^{\lambda \mathbf{s}_i} - \lambda\sum_{i=1}^n p_i\mathbf{s}_i
\end{aligned}
\tag{23}
$$

Based on mean field theory (Barabási, Albert, and Jeong 1999), there is:

$$
\log\frac{1}{n}\sum_{i=1}^n e^{\lambda \mathbf{s}_i} \approx \log\exp\left(\frac{1}{n}\sum_{i=1}^n \lambda\mathbf{s}_i\right) = \lambda\bar{s} \tag{24}
$$

Moreover, the softmax operation tends to the max value of $a_{i,j}$ (Qin et al. 2022), we have:

$$
\lambda\sum_{i=1}^n p_i\mathbf{s}_i \approx \lambda s_{\max} \tag{25}
$$

Therefore, the entropy in the attention mechanism can ultimately be approximated as follows:

$$
\mathcal{H} \approx \log n - \lambda(s_{\max} - \bar{s}) = \log n - \frac{1}{\sqrt{d}}(s_{\max} - \bar{s}) \tag{26}
$$

Assume that the form of the bias for penalizing attention is $f(|i-j|)$, where $f(|i-j|) > 0$, We denote the entropy after adding the bias as $\mathcal{H}'$, we have:

$$
\begin{aligned}
\mathcal{H}' &\approx \log n - \frac{1}{\sqrt{d}}((s_{\max} - a) - (\bar{s} - b)), \\
a &= f(|i_{s_{max}} - j_{s_{max}}|) > 0 \\
b &= \frac{\sum_{i=1}^n\sum_{j=1}^n f(|i-j|)}{nn}) > 0.
\end{aligned}
\tag{27}
$$

Based on Eq.(25), we have:

$$\mathcal{H} - \mathcal{H}' = \frac{1}{\sqrt{d}}(b - a) \tag{28}$$

The final result in Eq. (28) depends on the disparity corresponding to the coordinates of the maximum attention value s_{max} and the monotonicity of the bias function. Generally, the maximum attention value is likely to be concentrated near the diagonal (Press, Smith, and Lewis 2021; Chi et al. 2022), hence the disparity $|i - j|$ is expected to be smaller than the average $|i - j|$ within the matrix, i.e., $\frac{2}{3}n - \frac{2}{3}$. The kernelized bias function utilized in this paper is monotonically increasing. Therefore, the final result of Eq. (27) is highly likely to be greater than zero, i.e., $\mathbb{P}(\mathcal{H} - \mathcal{H}' > 0) \approx 1$.

Moreover, from the perspective of entropy invariance, we consider the impact of different attention score penalty strategies in the current KT on length generalization, we have:

$$\begin{aligned} &kernelized\ bias : \Delta_{kb} = \mathcal{H} - \mathcal{H}^{kb} = \frac{1}{\sqrt{d}}(b - a) \\ &\approx \tau_1 \log(1 + \tau_2(\frac{2}{3}n - \frac{2}{3})) \\ &linear\ bias : \Delta_{lb} = \mathcal{H} - \mathcal{H}^{lb} \approx 2^{-\frac{n}{H}}(\frac{2}{3}n - \frac{2}{3}) \\ &\Delta_{kb} - \Delta_{lb} = \mathcal{H}^{lb} - \mathcal{H}^{kb} \end{aligned} \tag{29}$$

By substituting $H = 8$, $n = 200$, $0 < \tau_1 \leq 1$, and $0 < \tau_2 \leq 2$, we can deduce that $\mathcal{H}^{lb} > \mathcal{H}^{kb}$. This implies that the entropy of $\mathcal{H}^{lb}$ is higher than that of $\mathcal{H}^{kb}$, meaning that $\mathcal{H}^{kb}$ is more concentrated in terms of attention distribution and is less likely to be influenced by extrapolated tokens, thereby dispersing attention less. This also explains why FlucKT performs better than FoLiBiKT and extraKT.

A3:Supplemental Experiments

1: Supplemental Experimental Settings Datasets: we introduce and compare each dataset in detail:

- **Algebra2005-2006 (AL2005)**: This dataset derives from the KDD Cup 2010 EDM Challenge, featuring interactions of 13-14-year-old students with Algebra questions. It contains detailed step-level responses to mathematical problems (Stamper and Pardos 2016). In our study, we create a unique identifier for each question by concatenating the problem name and step name.
- **Bridge to Algebra 2006-2007 (BD2006)**: The BD2006 dataset comprises mathematical problems derived from students' interactions with intelligent tutoring systems, as recorded in log files (Stamper and Pardos 2016). The construction of unique questions in BD2006 employs a format akin to that of AL2005.
- **NeurIPS2020 Education Challenge (NIPS34)**: This dataset is provided by the NeurIPS 2020 Education Challenge, specifically utilizing data from Tasks 3 and 4 to assess our models (Wang et al. 2020). It comprises students' responses to mathematics questions sourced from Eedi, a platform with millions of daily interactions globally. For KCs, we use the leaf nodes from the subject tree.

Among the many available KT datasets, only AL2005, BD2006, NIPS34, and AS2009 include both questions and their associated KCs. To analyze the effect of long content windows in attention-based KT models, datasets must have over 50% of sequences longer than 200 steps. This criterion is met only by AL2005, BD2006, and NIPS34, hence their selection for our research. We ensure reproducibility by rigorously following the data preprocessing steps outlined in (Liu et al. 2022b).

Baselines: we summarized the detailed information of baselines as follows:

- **DKT (Piech et al. 2015)**: DKT (Deep Knowledge Tracing) is the first model to integrate deep learning into the knowledge tracing (KT) task. Specifically, it employs Recurrent Neural Networks (RNNs) to model student learning processes and to estimate their mastery of questions and the associated KCs.
- **DKVMN (Zhang et al. 2017)**: DKVMN (Dynamic Key-Value Memory Networks) innovatively integrates dual memory structures for knowledge tracing: a static "key" memory for storing concepts and a dynamic "value" memory for updating mastery levels. This approach enhances prediction accuracy and uncovers underlying concepts, outperforming state-of-the-art models in various datasets.
- **SAKT (Pandey and Karypis 2019)**: SAKT (Self-Attentive Knowledge Tracing) uses a self-attention mechanism to address data sparsity in knowledge tracing, outperforming RNN-based models in both accuracy and efficiency by focusing on relevant past interactions.
- **SAINT (Choi et al. 2020)**: SAINT (Separated Self-AttentIve Neural Knowledge Tracing) introduces an encoder-decoder structure that separates exercise and response sequences for knowledge tracing. This design enhances the model's ability to capture complex relationships between exercises and student responses, leading to improved prediction accuracy.
- **AKT (Ghosh, Heffernan, and Lan 2020)**: AKT (Attentive Knowledge Tracing) enhances KT by integrating attention mechanisms with cognitive and psychometric models. It employs a novel monotonic attention mechanism and Rasch model-based embeddings to provide context-aware representations of student responses, improving both predictive performance and interpretability .
- **ATKT (Guo et al. 2021)**: ATKT (Adversarial Training-based Knowledge Tracing) enhances knowledge tracing by incorporating adversarial training to improve model robustness and generalization. By adding perturbations to interaction embeddings and using an attentive-LSTM backbone, ATKT significantly outperforms traditional models in various benchmark datasets.
- **LPKT (Shen et al. 2021)**: LPKT (Learning Process-consistent Knowledge Tracing) introduces a novel approach to knowledge tracing by modeling students' learning processes directly. It incorporates learning gains and forgetting effects to better capture students' evolving

knowledge states, achieving higher accuracy and interpretability compared to state-of-the-art methods.

- **SimpleKT (Liu et al. 2022a)**: SimpleKT employs a scaled dot-product attention mechanism to capture complex relationships between questions and their corresponding KCs. To account for individual differences among questions within the same KC, it defines a question-specific difficulty vector.
- **AT-DKT (Liu et al. 2023)**: AT-DKT improves upon the original DKT model by incorporating two auxiliary learning tasks: question tagging (QT) and individualized prior knowledge (IK) prediction. These tasks enhance student assessment by modeling question-KC relationships and estimating students' historical performance, leading to more accurate and robust knowledge tracing predictions across various datasets.
- **FoLiBiKT (Im et al. 2023)**: FoLiBiKT (Forgetting-aware Linear Bias for Knowledge Tracing) introduces a linear bias mechanism to account for forgetting behavior in attention-based knowledge tracing models. By decoupling question correlations from forgetting effects, FoLiBiKT improves prediction accuracy and robustness across various datasets, outperforming existing KT models.
- **DTransformer (Yin et al. 2023)**: DTransformer uses a dynamic memory-enhanced Transformer model for knowledge tracing, capturing temporal dependencies and evolving student knowledge states, leading to better prediction accuracy and understanding of learning processes.
- **extraKT (Li et al. 2024)**: extraKT enhances length generalization in knowledge tracing by using a length generalization module that penalizes attention scores with linearly decreasing biases. This model maintains performance stability across varying context window sizes, outperforming state-of-the-art models in AUC and accuracy.

Implementation Details: We adopt standardized experimental settings from pyKT (Liu et al. 2022b). Models are trained on student interaction sequences of length 200 and evaluated on sequences of lengths 200, 400, 600, 800, and 1000. Specifically: 1) A 5-fold cross-validation is employed, with 60% for training, 20% for validation, and 20% for testing. 2) Early stopping with a patience of 10 is applied during training. Models are optimized using Adam for up to 200 epochs per hyper-parameter combination, with Bayesian search for hyper-parameter tuning. Key hyper-parameters include embedding, hidden state, and prediction dimensions [64, 128, 256]; learning rate [1e-3, 1e-4, 1e-5]; dropout rate [0.05, 0.1, 0.3, 0.5]; and random seed [42, 3407]. Consistent with prior studies (Piech et al. 2015; Ghosh, Heffernan, and Lan 2020; Shen et al. 2021; Liu et al. 2022a; Im et al. 2023; Yin et al. 2023), we report the average AUC and ACC (accuracy) with standard deviations across 5-fold cross-validation for KT prediction evaluation.

2: Supplemental Experimental Results **AUC Results of Overall Performance:** Tables 5, 6, and 7 are the complete results of the AUC results of FlucKT on three datasets.

Model	AUC				
	Length of Interaction Sequences				
	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
DKVMN	0.8054±0.0011	0.8039±0.0014	0.8030±0.0016	0.8025±0.0017	0.8023±0.0018
GKT	0.8110±0.0009	0.8111±0.0009	0.8111±0.0009	0.8111±0.0009	0.8111±0.0009
SAKT	0.7899±0.0036	0.6743±0.0023	0.6691±0.0030	0.6677±0.0024	0.6666±0.0018
SAINT	0.7715±0.0018	0.6691±0.0110	0.6589±0.0021	0.6539±0.0017	0.6551±0.0016
AKT	0.8306±0.0019	0.8277±0.0030	0.8258±0.0038	0.8241±0.0045	0.8227±0.0051
ATKT	0.7995±0.0023	0.7816±0.0025	0.7641±0.0039	0.7523±0.0047	0.7446±0.0050
LPKT	0.8268±0.0004	0.8216±0.0019	0.8107±0.0104	0.7990±0.0181	0.7891±0.0197
SimpleKT	0.8210±0.0014	0.7808±0.0078	0.7763±0.0055	0.7535±0.0263	0.7655±0.0169
AT-DKT	0.8246±0.0019	0.8238±0.0019	0.8235±0.0019	0.8233±0.0020	0.8233±0.0020
FoLiBiKT	0.8310±0.0010	0.8288±0.0007	0.8272±0.0014	0.8256±0.0017	0.8242±0.0020
DTransformer	0.8188±0.0025	0.8156±0.0025	0.8137±0.0028	0.8123±0.0030	0.8112±0.0033
extraKT	0.8317±0.0021	0.8317±0.0020	0.8317±0.0019	0.8317±0.0019	0.8317±0.0019
FlucKT	**0.8376±0.0011**	**0.8370±0.0012**	**0.8365±0.0011**	**0.8360±0.0012**	**0.8358±0.0012**

Table 5: Performance comparisons in terms of AUC on AL2005 dataset.

Model	AUC				
	Length of Interaction Sequences				
	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
DKVMN	0.7983±0.0009	0.7956±0.0009	0.7936±0.0010	0.7925±0.0012	0.7919±0.0014
GKT	0.8046±0.0008	0.8047±0.0009	0.8047±0.0009	0.8047±0.0010	0.8047±0.0010
SAKT	0.7739±0.0015	0.7097±0.0056	0.7000±0.0042	0.6987±0.0035	0.6962±0.0044
SAINT	0.7791±0.0018	0.6847±0.0035	0.6816±0.0027	0.6692±0.0037	0.6697±0.0024
AKT	0.8208±0.0007	0.8187±0.0008	0.8168±0.0010	0.8155±0.0012	0.8144±0.0014
ATKT	0.7889±0.0008	0.7641±0.0028	0.7370±0.0041	0.7142±0.0042	0.6963±0.0040
LPKT	0.8056±0.0008	0.8014±0.0021	0.7965±0.0029	0.7939±0.0031	0.7923±0.0031
SimpleKT	0.8151±0.0006	0.7897±0.0046	0.7764±0.0124	0.7726±0.0090	0.7724±0.0088
AT-DKT	0.8104±0.0009	0.8098±0.0008	0.8095±0.0007	0.8092±0.0006	0.8089±0.0006
FoLiBiKT	0.8199±0.0008	0.8171±0.0007	0.8145±0.0011	0.8125±0.0016	0.8110±0.0020
DTransformer	0.8093±0.0009	0.8052±0.0020	0.8023±0.0029	0.8002±0.0035	0.7985±0.0039
extraKT	0.8247±0.0006	0.8246±0.0005	0.8246±0.0005	0.8245±0.0005	0.8245±0.0005
FlucKT	**0.8269±0.0005**	**0.8263±0.0007**	**0.8257±0.0007**	**0.8253±0.0007**	**0.8250±0.0007**

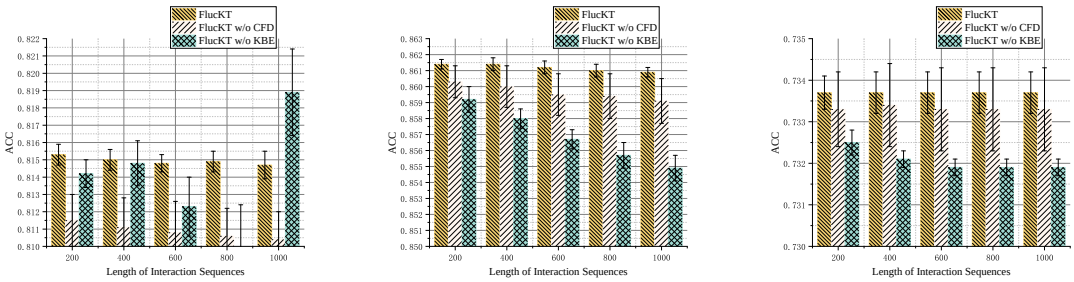
Table 6: Performance comparisons in terms of AUC on BD2006 dataset.

ACC Results of Overall Performance: Tables 8-10 present the results of the ACC resultss in the overall performance experiments. We observe that FlucKT achieved the best performance on both the AL2005 and BD2006 datasets. Although FlucKT did not achieve the highest ACC on the NIPS34 dataset, it still obtained the second-best result among all methods. Overall, the ACC results further demonstrate the effectiveness of FlucKT.

ACC Results of Ablation Study: Fig. 8 and Fig. 9 show the ACC results of the ablation study. Similar to the AUC results, the ACC results also confirm the effectiveness of the CFD and KBE components designed in FlucKT.

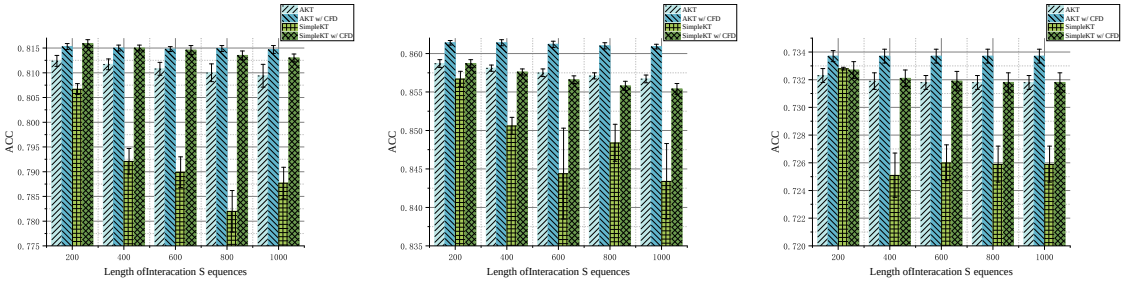
Case Study: We employed a case study to qualitatively visualize the prediction results of the model in specific sequences, with the color intensity representing the predicted probability. It can be observed that our cognitive feature decomposition layer indeed enhances the model's ability to predict certain fluctuation features. For example, the prediction performance for question IDs 47, 38, 36, and 48, 31, 26 in Fig. 10, and for question IDs 47, 31, 26, and 49, 31, 26 in Fig. 10. (b) is significantly improved.

Quantitative Experimental Results on More Datasets: Additionally, we performed additional quantitative experiments in two additional data sets, ASSISTments2015 and XES3G5M, setting the length of interaction sequences at 200. The ASSISTments2015 dataset comprises student responses from 100 skill builders with the highest number of student interactions, focusing on various mathematical skills. The XES3G5M dataset is collected from a real-world online math learning platform, containing 7,652 questions,



(a) AL2005 dataset. (b) BD2006 dataset. (c) NIPS34 dataset.

Figure 8: Ablation study results (FlucKT) in terms of ACC on three datasets.



(a) AL2005 dataset. (b) BD2006 dataset. (c) NIPS34 dataset.

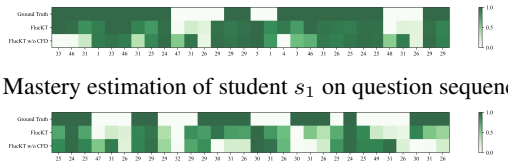
Figure 9: Ablation study results (AKT and SimpleKT) in terms of ACC on three datasets.

Model	AUC				
	Length of Interaction Sequences				
	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
DKVMN	0.7673±0.0004	0.7673±0.0004	0.7673±0.0004	0.7672±0.0004	0.7672±0.0004
GKT	0.7689±0.0024	0.7689±0.0025	0.7689±0.0025	0.7689±0.0025	0.7689±0.0025
SAKT	0.7525±0.0009	0.7331±0.0013	0.7329±0.0011	0.7330±0.0011	0.7330±0.0011
SAINT	0.7895±0.0009	0.7708±0.0009	0.7703±0.0012	0.7700±0.0012	0.7700±0.0012
AKT	0.8033±0.0003	0.8030±0.0004	0.8028±0.0004	0.8028±0.0004	0.8028±0.0004
ATKT	0.7665±0.0001	0.7630±0.0005	0.7620±0.0006	0.7619±0.0006	0.7619±0.0006
LPKT	0.8004±0.0003	0.7997±0.0005	0.7993±0.0006	0.7992±0.0007	0.7992±0.0006
SimpleKT	0.8035±0.0000	0.7952±0.0017	0.7961±0.0012	0.7960±0.0012	0.7960±0.0012
AT-DKT	0.7816±0.0002	0.7815±0.0002	0.7815±0.0002	0.7815±0.0002	0.7815±0.0002
FoLiBiKT	0.8032±0.0002	0.8029±0.0003	0.8028±0.0003	0.8028±0.0003	0.8028±0.0003
DTransformer	0.7994±0.0003	0.7988±0.0003	0.7985±0.0003	0.7985±0.0003	0.7985±0.0003
extraKT	0.8045±0.0003	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**
FlucKT	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**	**0.8047±0.0003**

Table 7: Performance comparisons in terms of AUC on NIPS34 dataset.

865 KC and 5,549,635 interactions from 18,066 students. As shown in Table 11, the experimental results indicate that FlucKT outperforms the baseline methods in both datasets, achieving higher AUC and ACC metrics. This quantitatively demonstrates FlucKT's strong generalization capabilities. **Complexity Analysis:** The complexity of FlucKT is mainly determined by its attention mechanism and the cognitive feature decomposition layer. For a sequence length L and hidden dimension D, the following observations can be made:

1. **Attention Mechanism:** The scaled dot-product attention operates with a complexity of $\mathcal{O}(L^2D)$, which is standard for attention-based architectures. Multi-head attention further scales this by the number of heads but operates in parallel.



(a) Mastery estimation of student s_1 on question sequence.

(b) Mastery estimation of student s_2 on question sequence.

Figure 10: Visualization of two students' knowledge mastery degree of questions over 30 steps

2. **Cognitive Feature Decomposition:** The causal convolution used in the decomposition layer has a complexity of $\mathcal{O}(L \cdot K \cdot D)$, where K is the kernel size. This step efficiently separates long-term trends from short-term fluctuations, avoiding the quadratic complexity of global operations.
3. **Kernelized Bias:** The kernelized bias attention mechanism introduces additional computation for position-based penalties, with an overhead linear in sequence length, resulting in $\mathcal{O}(LD)$.
4. **Overall Complexity:** Considering all components, the overall complexity of FlucKT per layer is approximately $\mathcal{O}(L^2D)$, with optimizations in convolution and kernel-based mechanisms making it effective for both short and long sequences.

Meanwhile, we compared FlucKT with several well-performing methods, namely DTransformer, extraKT, and

Model	ACC				
	Length of Interacation Sequences				
	200	400	600	800	1000
DKT	0.8097±0.0005	0.8098±0.0005	0.8098±0.0006	0.8098±0.0006	0.8098±0.0006
DKVMN	0.8027±0.0007	0.8025±0.0008	0.8023±0.0008	0.8022±0.0008	0.8022±0.0009
GKT	0.8088±0.0008	0.8087±0.0010	0.8088±0.0010	0.8088±0.0010	0.8088±0.0010
SAKT	0.7965±0.0019	0.7478±0.0016	0.7468±0.0026	0.7445±0.0017	0.7435±0.0020
SAINT	0.7755±0.0012	0.7355±0.0118	0.7424±0.0050	0.7291±0.0092	0.7324±0.0108
AKT	0.8124±0.0011	0.8117±0.0011	0.8108±0.0013	0.8100±0.0018	0.8094±0.0023
ATKT	0.7998±0.0019	0.7935±0.0026	0.7854±0.0049	0.7779±0.0072	0.7731±0.0090
LPKT	0.8154±0.0008	0.8123±0.0017	0.7970±0.0217	0.7746±0.0543	0.7613±0.0694
SimpleKT	0.8144±0.0008	0.8142±0.0008	0.8143±0.0008	0.8144±0.0007	0.8142±0.0007
AT-DKT	0.8138±0.0005	0.8131±0.0009	0.8123±0.0014	0.8117±0.0017	0.8113±0.0018
FoLiBiKT	0.8043±0.0021	0.8032±0.0021	0.8023±0.0023	0.8018±0.0023	0.8013±0.0026
DTransformer	0.8032±0.0002	0.8029±0.0003	0.8028±0.0003	0.8028±0.0003	0.8028±0.0003
extraKT	0.8110±0.0009	0.8109±0.0010	0.8108±0.0011	0.8108±0.0010	0.8109±0.0011
FlucKT	**0.8153±0.0006**	**0.8150±0.0006**	**0.8148±0.0005**	**0.8149±0.0006**	**0.8147±0.0008**

Table 8: Performance comparisons in terms of ACC on AL2005 dataset.

Model	ACC				
	Length of Interacation Sequences				
	200	400	600	800	1000
DKT	0.8553±0.0002	0.8553±0.0002	0.8552±0.0002	0.8552±0.0002	0.8552±0.0002
DKVMN	0.8545±0.0002	0.8540±0.0003	0.8537±0.0002	0.8535±0.0001	0.8534±0.0001
GKT	0.8511±0.0004	0.8555±0.0002	0.8556±0.0002	0.8556±0.0002	0.8556±0.0002
SAKT	0.8460±0.0004	0.8190±0.0030	0.8208±0.0030	0.8240±0.0008	0.8239±0.0009
SAINT	0.8445±0.0013	0.8396±0.0006	0.8373±0.0014	0.8396±0.0006	0.8396±0.0006
AKT	0.8587±0.0005	0.8581±0.0004	0.8575±0.0005	0.8571±0.0004	0.8567±0.0005
ATKT	0.8555±0.0002	0.8432±0.0020	0.8334±0.0033	0.8241±0.0043	0.8156±0.0058
LPKT	0.8547±0.0005	0.8539±0.0004	0.8524±0.0009	0.8507±0.0021	0.8495±0.0032
SimpleKT	0.8567±0.0010	0.8506±0.0011	0.8444±0.0059	0.8484±0.0024	0.8434±0.0049
AT-DKT	0.8560±0 0005	0.8558±0.0004	0.8557±0.0004	0.8556±0.0004	0.8555±0.0004
FoLiBiKT	0.8582±0.0007	0.8575±0.0003	0.8566±0.0001	0.8561±0.0004	0.8556±0.0004
DTransformer	0.8555±0.0007	0.8544±0.0007	0.8539±0.0010	0.8532±0.0010	0.8529±0.0010
extraKT	0.8605±0.0012	0.8605±0.0011	0.8605±0.0011	0.8605±0.0011	0.8605±0.0011
FlucKT	**0.8614±0.0003**	**0.8614±0.0004**	**0.8612±0.0004**	**0.8610±0.0004**	**0.8609±0.0003**

Table 9: Performance comparisons in terms of ACC on BD2006 dataset.

LPKT, in terms of runtime per epoch during the training phase and the number of epochs required for convergence. As shown in Table 12, FlucKT, while slower than extraKT, is significantly faster than DTransformer, which is also based on the Attention mechanism and the LPKT sequence modelLPKT. This is primarily because FlucKT does not require the generation of additional negative samples to construct contrastive loss. Additionally, the parallel computation capability of the Attention mechanism greatly enhances the training efficiency of the model. Therefore, we believe that FlucKT strikes a balance between performance and efficiency.

Model	ACC				
	Length of Interacation Sequences				
	200	400	600	800	1000
DKT	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004	0.7032±0.0004
DKVMN	0.7016±0.0005	0.7015±0.0005	0.7015±0.0005	0.7015±0.0005	0.7015±0.0005
GKT	0.7014±0.0028	0.7013±0.0029	0.7013±0.0029	0.7013±0.0029	0.7013±0.0029
SAKT	0.6884±0.0009	0.6741±0.0012	0.6739±0.0009	0.6740±0.0010	0.6740±0.0010
SAINT	0.7204±0.0009	0.7029±0.0012	0.7024±0.0012	0.7021±0.0012	0.7021±0.0012
AKT	0.7323±0.0005	0.7319±0.0006	0.7318±0.0005	0.7318±0.0005	0.7318±0.0005
ATKT	0.7013±0.0002	0.6988±0.0005	0.6980±0.0008	0.6980±0.0007	0.6980±0.0007
LPKT	0.7309±0.0006	0.7303±0.0012	0.7298±0.0015	0.7297±0.0016	0.7297±0.0015
SimpleKT	0.7328±0.0001	0.7251±0.0016	0.7260±0.0013	0.7259±0.0013	0.7259±0.0013
AT-DKT	0.7146±0.0002	0.7145±0.0003	0.7144±0.0003	0.7144±0.0003	0.7144±0.0003
FoLiBiKT	0.7323±0.0002	0.7320±0.0001	0.7319±0.0002	0.7319±0.0002	0.7319±0.0002
DTransformer	0.7295±0.0007	0.7289±0.0006	0.7286±0.0007	0.7286±0.0007	0.7286±0.0007
extraKT	**0.7340±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**	**0.7342±0.0004**
FlucKT	0.7337±0.0004	0.7337±0.0005	0.7337±0.0005	0.7337±0.0005	0.7337±0.0005

Table 10: Performance comparisons in terms of ACC on NIPS34 dataset.

Model	ASSISTment2015		XES3G5M	
	AUC	ACC	AUC	ACC
DKT	0.7271±0.0005	0.7503±0.0003	0.7852±0.0006	0.8173±0.0002
DKVMN	0.7254±0.0004	0.7500±0.0003	0.7792±0.0004	0.8155±0.0001
GKT	0.7258±0.0012	0.7504±0.0010	0.7727±0.0006	0.8135±0.0004
SAKT	0.7114±0.0003	0.7474±0.0002	0.7693±0.0008	0.8124±0.0002
SAINT	0.7026±0.0011	0.7438±0.0010	0.8074±0.0007	0.8177±0.0006
AKT	0.7281±0.0004	0.7521±0.0005	0.8207±0.0008	0.8273±0.0007
ATKT	0.7245±0.0007	0.7494±0.0002	0.7783±0.0004	0.8155±0.0001
LPKT	-	-	0.8163±0.0002	0.8264±0.0001
SimpleKT	0.7248±0.0005	0.7508±0.0004	0.8163±0.0006	0.8246±0.0005
AT-DKT	-	-	0.7932±0.0004	0.8198±0.0004
FoLiBiKT	0.7285±0.0003	**0.7526±0.0003**	0.8214±0.0007	0.8271±0.0006
DTransformer	0.7258±0.0005	0.7511±0.0006	0.8144±0.0006	0.8248±0.0004
extraKT	0.7291±0.0002	0.7516±0.0003	0.8200±0.0008	0.8263±0.0010
FlucKT	**0.7294±0.0003**	0.7519±0.0006	**0.8374±0.0341**	**0.8386±0.0167**

Table 11: Quantitative experimental results on ASSISTment2015 and XES3G5M datasets.

Runtime per epoch (s)	FlucKT	Dtransformer	extraKT	LPKT
AL2005	9.2	52.2	8.5	35.7
BD2006	17.4	105.3	16.2	90.6
NIPS34	16.3	101.4	15.3	91.2
Number of epochs for model convergence	FlucKT	Dtransformer	extraKT	LPKT
AL2005	14	17	14	15
BD2006	13	16	13	13
NIPS34	30	38	18	14

Table 12: Model training phase runtime