Cognitive Fluctuations Enhanced Attention Network for Knowledge Tracing

Anonymous submission

Abstract

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 the understanding of cognitive processes. Human cognition is characterized by two key features: long-term cognitive trends, which reflect 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. While 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[1], 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.

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 the 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 more accurately capture the dynamic sequential 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.

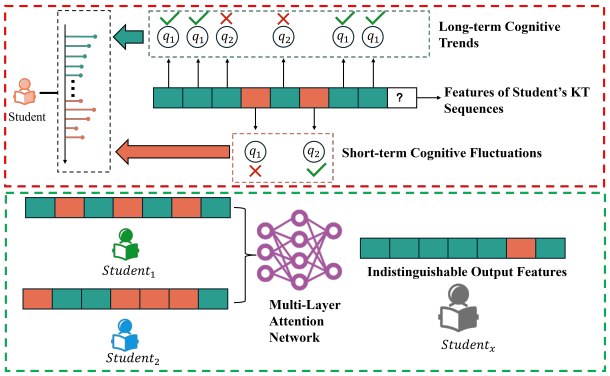


Figure 1: Illustration of cognitive trend and fluctuation features in learning data. Here, we represent the student's KT sequence features (Embeddings) with continuous blocks, where green denotes trend features and red indicates fluctuation features. As illustrated by the red dashed box, the trend features and fluctuation features within a student's KT sequence are intricately intertwined. As depicted by the blue dashed box, the distinct KT features of two students may become increasingly similar after passing through multiple layers of the attention network, making it difficult to distinguish between them.

Although attention mechanisms excel at capturing long-term cognitive trends in KT sequence data, they often fail to adequately address short-term cognitive fluctuations. This 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. Conversely, short-term cognitive fluctuations are more transient and often arise

[1]https://anonymous.4open.science/r/fluckt-FF6C

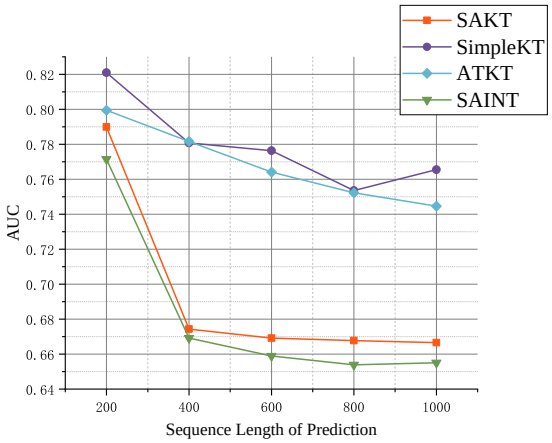


Figure 2: The AUC performance of KT models with different sequence lengths.

from random cognitive factors such as forgetting, fatigue-induced lapses in attention, carelessness, or guessing (Salthouse, Nesselroade, and Berish 2006; Pashler et al. 2009). Both long-term cognitive trends and short-term cognitive fluctuations influence students' overall knowledge state, 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 overly diffuse. When the length of the inference or test data exceeds the training data length (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 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: 1) First, we designed a decomposition-based layer to enhance the input features of the attention mechanism. This design uses causal convolution to decompose the attention inputs 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. 2) Second, we designed 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 the 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

Our work is related to KT and frequency domain learning; therefore, we provide a brief review of the related literature.

Knowledge Tracing

Knowledge tracing (KT) has advanced significantly with deep learning and innovative techniques. DKT (Deep Knowledge Tracing)(Piech et al. 2015) first introduced RNNs to model student learning. DKVMN (Dynamic Key-Value Memory Networks)(Zhang et al. 2017) added a dual memory structure for greater accuracy. GKT (Graph-based Knowledge Tracing)(Nakagawa, Iwasawa, and Matsuo 2019) utilized Graph Neural Networks (GNNs) to structure knowledge as a graph. SAKT (Self-Attentive Knowledge Tracing)(Pandey and Karypis 2019) applied self-attention to address data sparsity and improve prediction accuracy. SAINT (Separated Self-AttentIve Neural Knowledge Tracing)(Choi et al. 2020) used an encoder-decoder architecture to model exercise-response relationships effectively. LPKT (Learning Process-consistent Knowledge Tracing)(Shen et al. 2021) incorporated learning gains and forgetting effects for better predictions. Attention-based KT methods, like 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 (Yin et al. 2023) and extraKT (Li et al. 2024) further refined KT by improving temporal dependencies and length generalization.

However, existing attention-based KT models focus excessively on long-term cognitive trends in KT data, neglecting short-term cognitive fluctuations. This oversight leads to two main problems: first, it causes over-smoothing of the learned features, reducing the model's performance in KT prediction tasks; second, it results in scattered attention, impairing the model's performance in length generalization.

Frequency Domain Learning

Frequency domain learning (FDL) is a method used in signal processing and machine learning that transforms data from the time domain to the frequency domain through mathematical transformations such as the Fourier Transform (Baxes

1994; Peng, Sugiyama, and Mine 2022; Cheung et al. 2020). This approach allows for the analysis and processing of signals based on their frequency components, which often simplifies complex data and reveals patterns not easily detectable in the time domain. FDL 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, whereas low-frequency signals correspond to the overall 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).

FDL inspired us to consider the problems of existing attention-based KT models from the perspective of high-frequency and low-frequency signals. However, unlike time-series data, directly applying Fourier transform to convert KT data from the time domain to the frequency domain may cause information leakage, primarily due to the auto-regressive training paradigm prevalent in existing KT studies.

Preliminaries

Unless otherwise specified, we denote sets with copperplate uppercase letters (i.e., $\mathcal{A}$), matrices with bold uppercase letters (i.e., $\mathbf{A}$), and vectors with bold lowercase letters (i.e., $\mathbf{a}$).

In online learning systems, a learner's behavior is primarily composed of interaction records, which include a sequence of questions and the corresponding responses. For learner i at time step t, they will answer a question $q_t^i \in \mathbb{Q}$ drawn from a knowledge concept $c_t^i \in \mathbb{C}$, and receive a response $r_t^i \in \{0, 1\}$. Here, $r_t^i = 1$ indicates the learner answered the question correctly, while $r_t^i = 0$ 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 Problem: 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}.

Proposed Method

In this section, we detail the proposed short-term cognitive fluctuations enhanced attention network for knowledge tracing (FlucKT). As illustrated in Fig. 3, FlucKT starts with an embedding layer that encodes learners' sequential interactions into embeddings. After the embedding layer, we

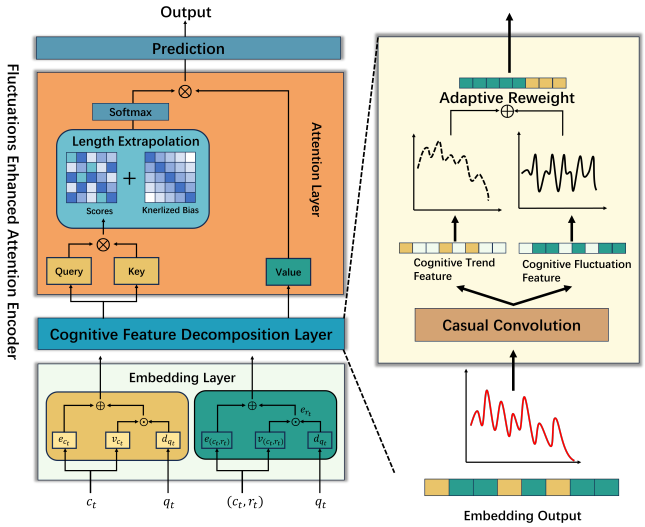


Figure 3: The overview of the proposed FlucKT framework.

improve the attention mechanism by incorporating short-term cognitive fluctuations in two ways: 1) We first use causal convolution to decompose the embeddings of interaction sequences into long-term cognitive trend features and short-term cognitive fluctuation features. These two features are then reorganized through adaptive weighted aggregation; 2) Within the attention layer, we introduce a kernelized bias into the attention scores to penalize long-term attention scores, thereby making the attention more focused. This ensures that the attention distribution remains concentrated during the testing phase, enhancing the extrapolation capability of the attention mechanism.

Embeddding Layer

Given that questions associated with the same knowledge components (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:

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

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.

Fluctuations Enhanced Attention Encoder

After efficiently representing the questions, KCs, and responses in KT, the next step is to capture the students'

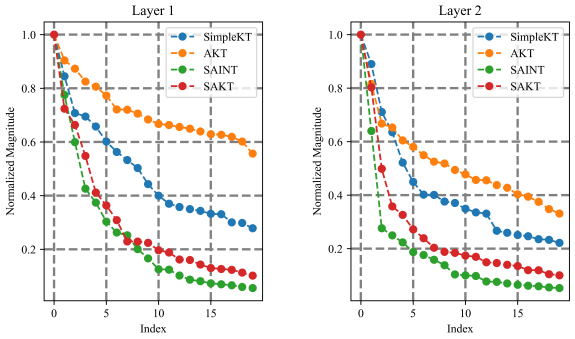


Figure 4: Visualization of the changes in the top 20 largest eigenvalues in the attention matrix as the number of layers increases (1− >2) in the attention-based KT model.

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 $\mathbf{A}\mathbf{h}_{l/s}$. 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 $\mathbf{h}_{high}$, we have:

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

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). 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. 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 FDL 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 extrapolation 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 extrapolation 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 (CPD) kernels.

In Eq.(3), if $a_{i,j}$ is an element of $\mathbf{A}$, we have

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

where $\epsilon = \sqrt{D}$. Our modified attention scores can be defined as:

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

where r_1 and r_2 are two learnable parameters that satisfy $0 < \tau_1 \leq 1$, and $0 < \tau_2 \leq 2$.

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:

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

where γ and δ denote Sigmoid and Relu functions. $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{W}_1$, $\mathbf{W}_2$ are trainable parameters and $\mathcal{L}$ represents the binary cross-entropy loss function.

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.

Among the many datasets available for knowledge tracing, only AL2005, BD2006, NIPS34, and AS2009 include both questions and their associated knowledge components (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 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). Due to space constraints, detailed descriptions of the baseline models are provided in Appendix A3.1.

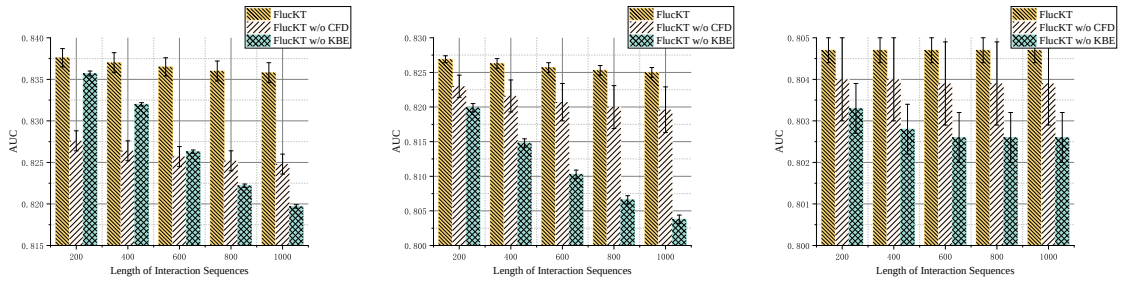
Implementation Details We follow standardized procedures for the experimental setup, primarily adopting the settings from pykt (Liu et al. 2022b). Specifically: 1) We employ a 5-fold cross-validation approach, where 60% of the data is used for training, 20% for validation, and 20% for testing. 2) During training, we utilize an early stopping strategy with a patience of 10. The models are trained using the Adam optimizer for up to 200 epochs for each hyper-parameter combination, and Bayesian search is used to determine the optimal hyper-parameters 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 (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 accuracy, along with their standard deviations, across the 5-fold cross-validation to evaluate the KT prediction performance.

Experiential Results

Overall Performance Tables 2-4 and Tables 5-7 present the overall performance of various models, including our proposed FlucKT model, on the AL2005, BD2006, and NIPS34 datasets in terms of AUC and ACC, with ACC results detailed in the Appendix (See Appendix A3.2). The best AUC and ACC values are highlighted in bold, while the second-best are underlined. From these tables, we observe that: 1) On 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

Table 1: Statistics of the datasets.

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%



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

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

Table 2: Performance comparisons in terms of AUC on AL2005 dataset.

Model	AUC				
	Length of Interacation 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**

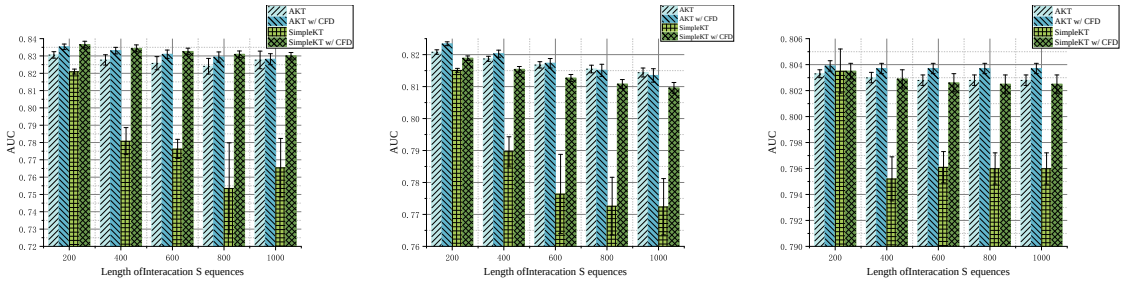
Table 3: Performance comparisons in terms of AUC on BD2006 dataset.

Model	AUC				
	Length of Interacation 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**

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 across 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.

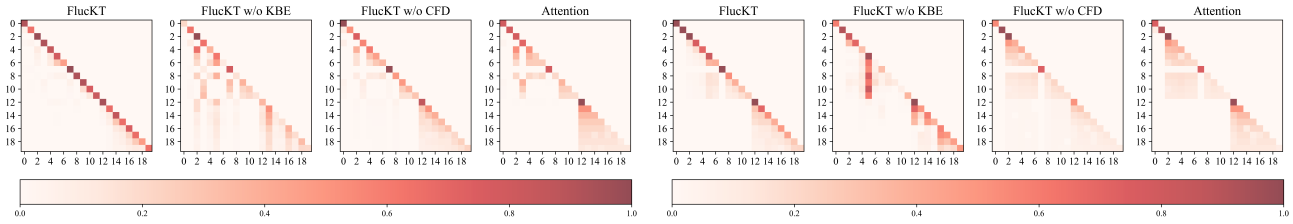
While the AUC increase in Tables 2-4 is less than 1%, this improvement is significant at a context window size of 200. Recent benchmark studies indicate that many reported performance gains are unreliable due to flawed evaluation processes, with only a 3.5% improvement in overall KT prediction performance since 2015. In our study, we rigorously followed the evaluation process proposed by pykt (Liu et al. 2022b) and conducted a thorough hyperparameter search for each baseline.

Ablation Study We systematically examine the effect of two key components in our FlucKT by constructing two model variants: the CFD represents the cognitive feature decomposition layer and the KBE denotes the kernelized bias enhanced attention scores. We conduct ablation study on three datasets and we only show the resluts in terms of AUC on three datasets (The ACC results of the ablation study are provided in the Appendix A3.2). Based on our ablation study results (Figs. 5- 6 and Figs. 8- 9), we have the following observations: 1) From the ablation results of AUC and ACC across the three datasets, it is evident that the performance of FlucKT decreases when either CFD or KBE is removed. This demonstrates that both CFD and KBE are indispensable components for FlucKT. 2) The impact of CFD and KBE varies across different datasets: CFD has a more pronounced effect on the AL2005 dataset, whereas KBE shows



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

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



(a) Averaged attention scores on the 1st layer. (b) Averaged attention scores on the 2nd layer.

Figure 7: Visualization of frequency rescaler layer (FRL) and kernelized bias enhanced attention scores (KBE) impact on attention scores.

Table 4: Performance comparisons in terms of AUC on NIPS34 dataset.

Model	AUC				
	Length of Interacation 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**

a more significant influence on the other two datasets. 3) The CFD also exhibits notable effects on other attention-based KT models. For instance, AKT with CFD (AKT w/ CFD) and SimpleKT with CFD (SimpleKT w/ CFD) both show varying degrees of improvement in AUC (as shown in Fig. 6) and ACC (as shown in Fig. 9) across the three datasets. This is particularly evident for SimpleKT, which lacks any length generation capabilities. After incorporating CFD, its prediction performance in the 200-1000 step range improves significantly. This further proves that the design of cognitive feature decomposition layer not only enhances the performance of attention-based KT models but also strengthens their length generation capabilities.

Visualization To qualitatively analyze the impact of the two improvements in FlucKT (i.e., CFD and KBE), we visualized the attention scores, as shown in Fig. 7. 'Attention' denotes the FlucKT model after removing the CFD and KBE modules. We observed that: 1) FlucKT causes the distribution of attention scores to be more focused and more attentive to short-term features; 2) CFD indeed enhances attention between certain proximal time steps. By comparing 'FlucKT w/o KBE' in (a) with 'Attention', we observe that the correlations between certain time steps are strengthened, such as the sequence after index 14. This trend becomes more apparent with an increase in layers, as shown in (b); 3) 'FlucKT w/o CFD' demonstrates that KBE penalizes long-term attention scores to some extent, making the attention more concentrated; 4) It is observable that, with the increase in layers, FlucKT not only captures local features but also captures mesoscopic features that lie between local and global characteristics to a certain extent. Additionally, we conducted a case study to qualitatively compare the performance of FlucKT and its variants. Detailed experimental results are provided in Appendix A3.2.

Conclusion

This paper presents FlucKT, an enhanced attention-based knowledge tracing (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 advancements contribute to the development of more effective personalized education systems.

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 seventh ACM conference on learning@ 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 & 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 ACM Web Conference 2023*, 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 *The Eleventh 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.

Nakagawa, H.; Iwasawa, Y.; and Matsuo, Y. 2019. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *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 *12th International Conference on Educational Data Mining, EDM 2019*, 384–389.

Pashler, H.; Cepeda, N.; Lindsey, R. V.; Vul, E.; and Mozer, M. C. 2009. Predicting the optimal spacing of study: A multiscale context model of memory. *Advances in neural information processing systems*, 22.

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 *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 *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 & 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.

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 ACM Web Conference 2023*, 855–864.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, 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 conference on World Wide Web*, 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 *International conference on machine learning*, 27268–27286.

A1:Proof of Theorem 1

Theorem 1 (Self-Attention is a low-pass filter): Let $\mathbf{A}$ be the self-attention score matrix. Then A inherently acts as a low-pass filter. For all $\mathbf{h}_{low}$ and $\mathbf{h}_{high}$, in other words,

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

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:

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

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, $\lim_{l\to\infty}|\cos\left(\langle \mathbf{A}^{l+1}\mathbf{h}, \mathbf{p}_1\rangle\right)| = 1$.

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

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 extrapolation 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 extrapolation. 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 $0 \leq i \leq n$ and $0 \leq j \leq n$.

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.(26), 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. (28) 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 extrapolation, 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 knowledge components (KCs), we use the leaf nodes from the subject tree.

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 knowledge components (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 knowledge components (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 extrapolation in knowledge tracing by using a length extrapolation 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.

2: Supplemental Experimental Results

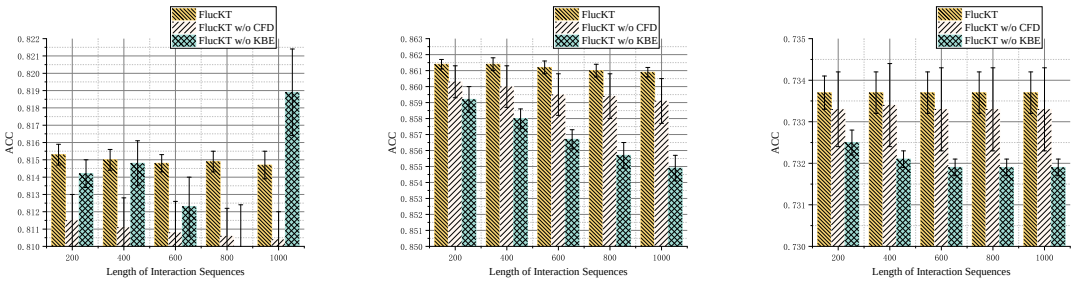
ACC Results of Overall Performance: Tables 5-7 present the results of the ACC resultss in the overall performance experiments. We observe that FlucKT achieved the best performance on both the AL2005 and BD2006 datasets. Although FlucKT did not achieve the highest ACC on the NIPS34 dataset, it still obtained the second-best result among all methods. Overall, the ACC results further demonstrate the effectiveness of FlucKT.

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

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	.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**

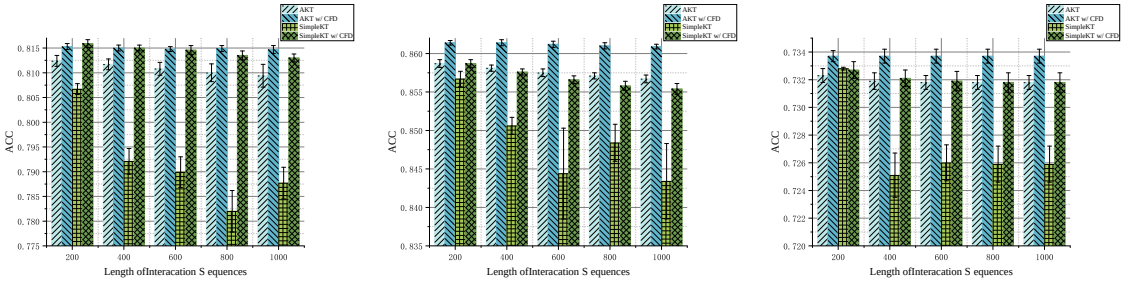
Table 6: Performance comparisons in terms of ACC on BD2006 dataset.

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**



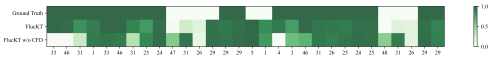
(a) AL2005 dataset. (b) BD2006 dataset. (c) NIPS34 dataset.

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

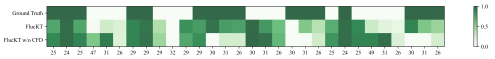


(a) AL2005 dataset. (b) BD2006 dataset. (c) NIPS34 dataset.

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



(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

ACC Results of Ablation Study: Fig. 8 and Fig. 9 show the ACC results from 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 model's prediction results on specific sequences, with 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 instance, the prediction performance for question IDs 47, 38, 36, and 48, 31, 26 in Fig. 10.(a), and for question IDs 47, 31, 26, and 49, 31, 26 in Fig. 10.(b) is notably improved.

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

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