Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

LLM4KT: Enhancing Knowledge Tracing via Large Language Models

Shuaishuai Zu
Renmin University of China
Biao Qin
qinbiao@ruc.edu.cn
Renmin University of China
Jihao Zhao
Renmin University of China
Shuhao Guan
University College Dublin

Research Article

Keywords: Knowledge tracing, Large language models, Semantic compression, Token reweighting

Posted Date: May 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9653429/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

LLM4KT: Enhancing Knowledge Tracing via Large Language Models

Shuaishuai Zu[1], Biao Qin[1*], Jihao Zhao[1], Shuhao Guan[2]

[1*]Renmin University of China, Beijing, Beijing, China.
[2]University College Dublin, Dublin, Ireland.

*Corresponding author(s). E-mail(s): qinbiao@ruc.edu.cn;
Contributing authors: zushuaishuai@gmail.com; zhaojihao@ruc.edu.cn;
shuhao.guan@ucdconnect.ie;

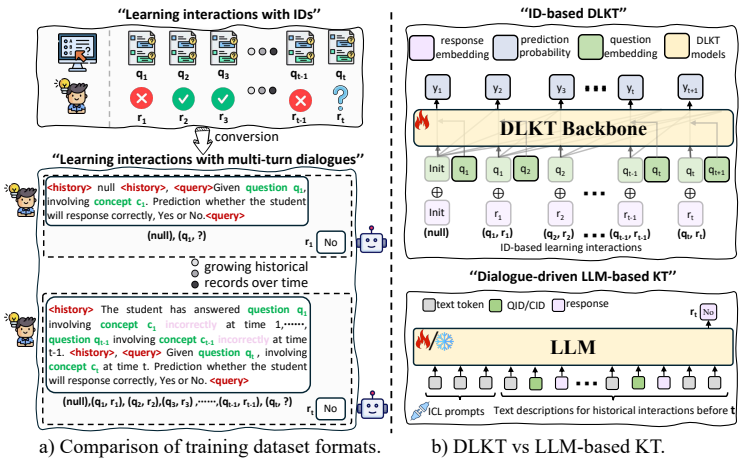
Abstract

Knowledge tracing (KT), a critical task in educational technology, aims to model students' knowledge states and predict their future performance over time. Large Language Models (LLMs) have recently emerged as promising tools in various fields, prompting researchers to explore their potential in KT. However, these attempts, whether employing fine-tuning or in-context learning, demonstrate limited performance gains. The main bottleneck lies in understanding lengthy text descriptions that describe students' ID-based interaction sequences. In this paper, to further unlock the potential of LLMs for KT, we propose LLM4KT, a novel hierarchical framework with an Inter LLM for semantic compression and a Tracer LLM for temporal modeling. First, textual descriptions of each interaction are fed into the Inter LLM independently to be compressed into a semantic-rich token embedding. During compression, we propose a novel token reweighting mechanism that dynamically identifies and reweights tokens based on gradient analysis for preserving critical information. Subsequently, Tracer LLM processes these condensed token embeddings to trace students' evolving knowledge states. In addition to the primary performance prediction task, we propose an auxiliary question recall task to explicitly augment the model's memory of prior interactions, thereby facilitating predictive performance. To support both tasks, we design task-specific prompt templates and employ the Inter LLM as two functional roles—Query LLM and Recall LLM—to compress respective task information for upcoming questions. Experiments on five benchmark datasets demonstrate that LLM4KT achieves new state-of-the-art performance.

Keywords: Knowledge tracing, Large language models, Semantic compression, Token reweighting

1 Introduction

Knowledge tracing (KT) [28] plays a critical role in intelligent tutoring systems and adaptive online learning platforms by monitoring students' knowledge acquisition processes and predicting their future performance. Accurate KT enables personalized learning experiences, helping educators identify knowledge gaps and recommend appropriate learning resources to students [2]. Existing KT methods typically rely on ID-based learning records, treating each interaction as discrete entity IDs of corresponding questions and concepts [38] and mapping them to learnable embedding vectors. With the development of deep learning, a variety of deep neural networks have been adopted to model temporal dependencies among interactions [11, 35, 39], facilitating deep learning-based KT (DLKT) models to achieve remarkable improvements.



a) Comparison of training dataset formats. b) DLKT vs LLM-based KT.

Fig. 1 Comparison of Dialogue-driven LLM-based KT methods and ID-based DLKT methods.

Recently, the emergence of large language models (LLMs) [3], which exhibits powerful text understanding and reasoning capabilities [38], has marked a paradigm shift in scale and capability relative to previous deep learning techniques for KT. Existing LLM-based attempts primarily rely on either in-context learning [13] or fine-tuning to adapt LLMs for the KT task [12, 14]. Both methodologies adopt a dialogue-driven format, where students' ID-based interactions are converted into multi-turn textual conversational exchanges. However, despite this seeming compatibility with LLMs' text-in format, these LLM-based KT methods only marginally outperform ID-based DLKT methods. As shown in Figure 1 a), students progressively engage with learning platforms, accumulating a growing sequence of question-solving interactions. The textual conversion of this ID-based sequence necessitates splitting the complete interaction sequence into multi-turn dialogues, inevitably resulting in increasingly lengthy textual descriptions as historical sequences grow longer. Subsequently, as illustrated in

Figure 1 b), LLM-based KT methods iteratively utilize these converted textual descriptions as inputs to predict students' future performance, which contrasts with previous ID-based DLKT methods that directly process ID-based sequences. Therefore, two major challenges emerge: 1) Utilizing the continuously growing text descriptions of learning behaviors as inputs increases computational cost, as the self-attention mechanism underpinning their architecture scales quadratically with sequence length, making accurate long-term tracking impractical; 2) Lengthy text descriptions introduce redundant information, potentially diluting key task-relevant information and increasing the difficulty for LLMs to understand students' learning behaviors [16].

To address these challenges, we propose a novel hierarchical framework, termed **LLM4KT**[1], that solves the KT task with decomposed semantic compression and temporal dependency modeling layers. First, an Inter LLM serves as a semantic feature extractor and compresses textual descriptions of each interaction independently into semantic-rich token embeddings. Crucially, we develop a novel token reweighting mechanism based on gradient analysis that adaptively identifies and preserves critical information during the compression process. Next, a Tracer LLM processes these condensed token embeddings to model students' evolving knowledge states over time. By transforming lengthy textual descriptions of historical interactions into compressed token embeddings through the Inter LLM, the Tracer LLM achieves input lengths comparable to previous ID-based KT methods, thereby leveraging rich semantic information while reducing computational burden compared to directly processing lengthy text inputs. Then, beyond the primary performance prediction task, we introduce an auxiliary question recall task that enables the model to determine whether students have previously encountered questions involving the same knowledge concepts as the upcoming question. This task enhances the Tracer LLM's memory of historical interactions, which contain valuable contextual signals to promote performance prediction. Finally, to facilitate the understanding of both tasks, we design task-specific prompt templates to describe upcoming questions and use the Inter LLM to function as Query LLM and Recall LLM, compressing respective task information from these textual descriptions into token embeddings. The main contributions are as follows:

- We propose LLM4KT, which employs LLMs to enhance the KT task by adopting a hierarchical framework with an Inter LLM for interaction compression and a Tracer LLM for tracking students' knowledge states.
- We design a token reweighting mechanism for Inter LLM that dynamically reweights tokens based on gradient analysis, producing condensed token embeddings that preserve critical semantic information.
- We propose an auxiliary question recall task that enhances the Tracer LLM's memory of students' prior exposure to knowledge concepts, which provides valuable historical signals that facilitate the primary task.
- Extensive experiments on five benchmark datasets demonstrate that our LLM4KT-4.6B achieves new state-of-the-art results.

[1] The source code is provided in the supplementary file.

2 Related Work

Deep Learning-based Knowledge Tracing. Existing DLKT methods can be broadly categorized based on their underlying architectures: (1) RNN-based models [21, 26], particularly those utilizing LSTM structures [36] such as DKT [26], which sequentially process interactions to update knowledge state representations; (2) Transformer-based models [29] including FoLiBiKT [15] and DTransformer [35] that leverage modified attention mechanisms and inject modulated attention bias to alleviate the long-term forgetting problem; (3) GNN-based models [22, 39] that represent knowledge concepts and interactions as graph nodes to capture both structural relationships among concepts and temporal dependencies. (4) Hybrid architectures [5, 37] and other neural network variants [1, 30] to address the problem.

LLM-based Knowledge Tracing. Recent researchers have explored integrating LLMs with KT through different attempts. These methods can be broadly categorized into two main strategies: in-context learning (ICL) [17] based methods and fine-tuning based methods. ICL-based approaches leverage carefully crafted prompts to guide pre-trained LLMs in understanding students' learning behaviors and predicting their future performance without parameter optimization. For example, LOKT [13] introduces textual categorical option weights to encode interaction histories efficiently, achieving strong performance under strict token constraints in zero-shot settings. Fine-tuning-based methods, in contrast, adapt pre-trained language models through task-specific training on KT datasets. For example, LKT [14] and CLST [12] reformulate the KT task as a natural language processing task, converting ID-based interaction sequences into multi-turn dialogue formats and fine-tuning the LLMs on these formatted datasets. However, despite the formatted inputs are compatible with LLMs' text processing capabilities, these dialogue-driven LLM-based KT methods only achieve marginal performance improvements over previous ID-based DLKT approaches, primarily due to the lengthy textual inputs that grow with the learning process. The self-attention mechanism underpinning LLMs incurs substantial computational costs, particularly for long context processing [16]. Implementing preliminary context compression processes [6, 16, 32] is an effective approach to leverage LLM's powerful semantic understanding capability while addressing the computational challenge. Inspired by this, we introduce a novel hierarchical framework that fundamentally differs from previous approaches by decomposing semantic compression and temporal dependency modeling into two separate layers, thereby alleviate computational costs.

3 Method

In this section, we first provide the problem formulation, then present an overview of our LLM4KT framework, followed by detailed introductions of each component and the training objectives.

3.1 Problem Formulation

Given an interactions sequence $\mathcal{S}_{<t} = \{(q_1, c_1, r_1), ..., (q_{t-1}, c_{t-1}, r_{t-1})\}$ and an upcoming question q_t, the goal of KT is to predict the probability that the student will

answer q_t correctly:

$$p(r_t = 1|q_t, c_t, \mathcal{S}_{<t}), \tag{1}$$

where q_t represents the question encountered at time step t, and $r_t \in \{0, 1\}$ denotes the student's response correctness (0 for incorrect, 1 for correct). Each question q_t is associated with a knowledge concept c_t being assessed. Additionally, we introduce a question recall task that predicts whether the student has previously encountered questions involving the same concept as q_t:

$$p(m_t = 1|q_t, c_t, \mathcal{S}_{<t}), \tag{2}$$

where m_t equals 1 if any question involving concept c_t appears in $\mathcal{S}_{<t}$, and 0 otherwise. For the question recall task, the ground truth labels $\hat{m}_t$ are obtained from the student's interaction records in advance. Specifically, for each time step t and the question q_t involving concept c_t, we set $\hat{m}_t = 1$ if the student has previously encountered any question involving the same concept c_t in their interaction records $\mathcal{S}_{<t}$, and $\hat{m}_t = 0$ otherwise. Formally:

$$\hat{m}_t = \begin{cases} 1, & \text{if } \exists j < t \text{ such that } c_j = c_t \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

3.2 LLM4KT Framework Overview

Figure 2 illustrates the overall architecture of LLM4KT, which adopts a two-layer hierarchical design that first extracts semantic features from textual descriptions and then performs temporal dependency modeling: (1) An Inter LLM compresses individual interaction text descriptions into compact token embeddings; (2) A token reweighting mechanism is designed to recognize and preserve critical information during compression; (3) A Tracer LLM utilizes these condensed token embeddings as inputs to track students' evolving knowledge states over time; (4) The Inter LLM functions as both Query LLM and Recall LLM, using tailored task-specific prompt templates to describe upcoming questions; and (5) Feature Aligners bridge different LLM components by ensuring dimensional compatibility and semantic feature alignment.

3.2.1 Inter LLM

Given that LLMs have demonstrated excellent performance in context comprehension, we utilize Inter LLM as a semantic feature extractor [27]. Inspired by [23], a special token [INTER] is added at the end of each interaction's textual description. The interaction prompt template is shown in Figure 2 c). It is worth noting that when $t = 0$, i.e., when the student has not yet encountered any questions, we define a task instruction prompt as the input to initialize the learning process. Furthermore, for time step $t - 1$, the Inter LLM takes a sequence of textual words describing the interaction record $(q_{t-1}, c_{t-1}, r_{t-1})$ as input: "question QID= $\{q_{t-1}\}$ involving concept CID= $\{c_{t-1}\}$ at time step t-1 {correctly/incorrectly}, [INTER]", where "correctly" if

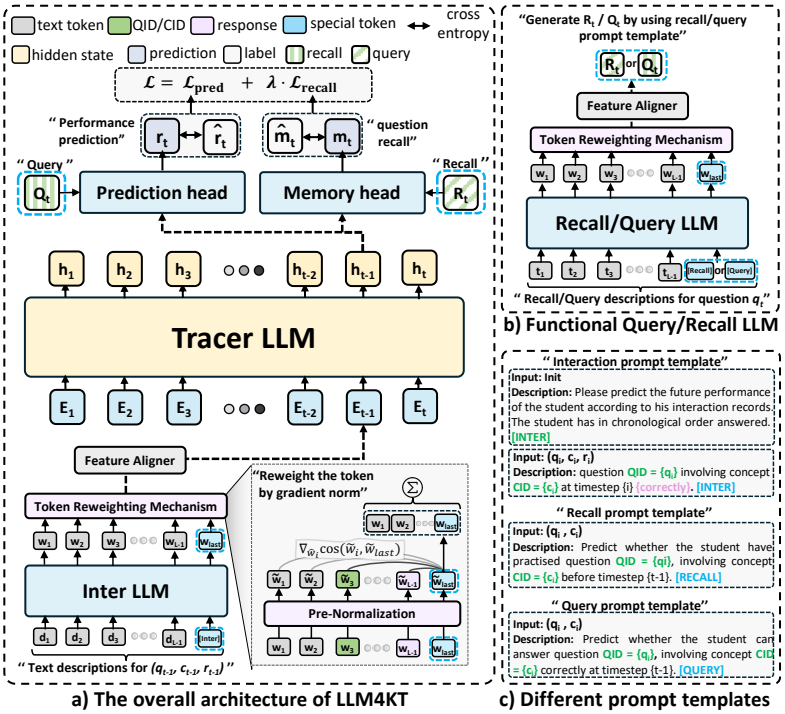


Fig. 2 LLM4KT Framework.

$r_{t-1} = 1$ and "incorrectly" if $r_{t-1} = 0$. This textual description is then tokenized into text tokens $\{d_1, d_2, d_3, \cdots, d_m, [\text{INTER}]\}$. Finally, we extract the token embedding at the position of the special token "[INTER]" from the last hidden states of Inter LLM as the semantic representation of the interaction record $(q_{t-1}, c_{t-1}, r_{t-1})$.

$$\mathbf{E}_{t-1}^{\text{inter}} = \text{Inter LLM}(d_1, d_2, d_3, \cdots, d_{L-1}, [\text{INTER}]), \tag{4}$$

where $\mathbf{E}_{t-1}^{\text{inter}} = \{w_1, w_2, w_3, \cdots, w_{L-1}, w_{\text{last}}\}$ denotes the last hidden states of Inter LLM, $w_i \in \mathbb{R}^{d_I}$ represents the token embedding corresponding to token d_i.

3.2.2 Token Reweighting Mechanism

In order to alleviate the potential dilution of critical information among less relevant tokens and ensure that the key information is preserved during prompt compression processes, we design a novel token reweighting mechanism. Instead of directly extracting the hidden state at the position of the special token, our mechanism further dynamically identifies and emphasizes tokens that contribute most stably to the subsequent temporal modeling in Tracer LLM. The core insight is that tokens with smaller gradient norms when computing cosine similarity with the special token exhibit greater stability and should contribute more to the final representation. Specifically,

for each token embedding w_i (including the special token embedding w_{last}) generated from textual description of record at timestamp i, we first normalize these token embeddings and then compute the gradient of the cosine similarity $c_i = \cos(\tilde{w}_i, \tilde{w}_{\text{last}})$ with respect to w_i:

$$\tilde{w}_i = \frac{w_i}{\|w_i\|}, \quad \tilde{w}_{\text{last}} = \frac{w_{\text{last}}}{\|w_{\text{last}}\|}, \tag{5}$$

where the gradient is computed as:

$$\begin{aligned} \nabla_{\tilde{w}_i} \cos(\tilde{w}_i, \tilde{w}_{\text{last}}) &= \frac{\tilde{w}_{\text{last}}}{\|\tilde{w}_i\|} - \frac{\tilde{w}_i \cdot \tilde{w}_{\text{last}}}{\|\tilde{w}_i\|} \tilde{w}_i \\ &= \frac{1}{\|\tilde{w}_i\|} (\tilde{w}_{\text{last}} - cos(\tilde{w}_i, \tilde{w}_{\text{last}}) \tilde{w}_i). \end{aligned} \tag{6}$$

Since we have normalized w_i ($\|\tilde{w}_i\| = 1$), the gradient norm g_i is then computed as:

$$\begin{aligned} g_i &= \|\nabla_{\tilde{w}_i} \cos(\tilde{w}_i, \tilde{w}_{\text{last}})\| \\ &= \frac{\tilde{w}_{\text{last}}^2 + cos^2(\tilde{w}_i, \tilde{w}_{\text{last}}) \tilde{w}_i^2 - 2cos(\tilde{w}_i, \tilde{w}_{\text{last}}) \tilde{w}_i \tilde{w}_{\text{last}}}{\|\tilde{w}_i\|^2} \\ &= 1 - cos(\tilde{w}_i, \tilde{w}_{\text{last}})^2. \end{aligned} \tag{7}$$

A smaller gradient norm g_i indicates that the cosine similarity between $\tilde{w}_i$ and special token $\tilde{w}_{\text{last}}$ is more stable under perturbations, suggesting that this token has a more consistent and reliable contribution to the subsequent tasks. To transform these gradient norms into meaningful weights, we apply a negative transformation followed by softmax normalization:

$$\alpha_i = \frac{\exp(-g_i)}{\sum_{j=1}^{m} \exp(-g_j)}. \tag{8}$$

The negative sign ensures that tokens with smaller gradient norms (higher stability) receive larger weights in the final weighted representation. The weighted embedding representation is then computed as:

$$\begin{aligned} \mathbf{E}_{t-1}^{\text{weight}} &= \text{TRM}(\mathbf{E}^{\text{inter}}) = \sum_{i=1}^{L-1} \alpha_i w_i + w_{last}, \\ \mathbf{E}_{t-1} &= \text{Aligner}(\mathbf{E}_{t-1}^{\text{weight}}), \end{aligned} \tag{9}$$

where $\text{TRM}(\cdot)$ denotes the token re-weighting mechanism, $\mathbf{E}_{t-1}^{\text{weight}}$ represents the weighted embedding of text descriptions of interaction $(q_{t-1}, c_{t-1}, r_{t-1})$, $\text{Aligner}(\cdot)$ denotes a feature aligner network. After obtaining the weighted embedding, we employ an $\text{Aligner}(\cdot)$ to transform $\mathbf{E}_{t-1}^{\text{weight}} \in \mathbb{R}^{1 \times d_I}$ into a dimension-compatible embedding representation $\mathbf{E}_{t-1} \in \mathbb{R}^{1 \times d_T}$ for the subsequent Tracer LLM. Importantly, our token

reweighting mechanism can reduce the information entropy of the token contribution distribution. This entropy reduction indicates decreased uncertainty and enhanced preservation of critical information during the compression process. We provide the proof below. We formalize the entropy reduction property as follows:

Theorem 1 *Let* $\tilde{\mathbf{c}} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_L\}$ *be the normalized distribution where* $\tilde{c}_i = \frac{|c_i|}{\sum_{j=1}^{L} |c_j|}$, *and let* $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_L\}$ *be the reweighted distribution where* $\alpha_i = \frac{\exp(-g_i)}{\sum_{j=1}^{L} \exp(-g_j)}$ *with* $g_i = 1 - c_i^2$. *Then the Shannon entropy of the reweighted distribution is bounded by that of the normalized cosine similarity distribution:*

$$H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{c}}), \tag{10}$$

where $H(\cdot)$ *denotes the Shannon entropy. Equality holds if and only if all* c_i *are equal.*

Proof We provide the proof of **Theorem 1**, which demonstrates that the token reweighting mechanism reduces the information entropy of the token contribution distribution. Consider the token reweighting process in our Inter LLM, where we transform the original cosine similarity distribution $\mathbf{c} = \{c_1, c_2, \ldots, c_L\}$ to the final weight distribution $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_L\}$ through the following steps:

$$c_i = \cos(\tilde{w}_i, \tilde{w}_{\text{last}}) \in [-1, 1], \tag{11}$$

$$g_i = 1 - c_i^2 \in [0, 1], \tag{12}$$

$$\alpha_i = \frac{\exp(-g_i)}{\sum_{j=1}^{L} \exp(-g_j)} = \frac{\exp(c_i^2 - 1)}{\sum_{j=1}^{L} \exp(c_j^2 - 1)}. \tag{13}$$

We define the normalized cosine similarity distribution as $\tilde{c}_i = \frac{|c_i|}{\sum_{j=1}^{L} |c_j|}$ and analyze the Shannon entropy relationship: $H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{c}})$.

We prove **Theorem 1** by decomposing the transformation $\mathbf{c} \to \boldsymbol{\alpha}$ into two stages: $\mathbf{c} \to \mathbf{g} \to \boldsymbol{\alpha}$. Following the notation, we further define the normalized gradient distribution as $\tilde{g}_i = \frac{g_i}{\sum_{j=1}^{L} g_j}$. We first analyze the entropy relationship between the original cosine similarity distribution and the final weight distribution. Our analysis demonstrates that the token reweighting mechanism reduces distributional entropy:

$$H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{c}}), \tag{14}$$

where equality holds if and only if all cosine similarities c_i are equal. This entropy reduction indicates decreased uncertainty and enhanced preservation of critical information.

We then establish the entropy reduction by decomposing the transformation $\mathbf{c} \to \boldsymbol{\alpha}$ into two stages: $\mathbf{c} \to \mathbf{g} \to \boldsymbol{\alpha}$.

Stage 1: Cosine Similarity to Gradient Norm ***($c_i \to g_i$)***

The transformation $f(c) = 1 - c^2$ exhibits the following properties:

- Symmetry: $f(-c) = f(c)$, causing sign information loss;
- Non-injectivity: Multiple c_i values map to identical g_i values.

By the data processing inequality [4], which states that deterministic transformations cannot increase information content, for any deterministic function f:

$$H(f(X)) \leq H(X). \tag{15}$$

Therefore: $H(\tilde{\mathbf{g}}) \leq H(\tilde{\mathbf{c}})$, where $\tilde{\mathbf{g}}$ is the normalized gradient distribution.

Stage 2: Gradient Norm to Weight ($g_i \to \alpha_i$)

The exponential transformation $\alpha_i = \frac{\exp(-g_i)}{\sum_j \exp(-g_j)}$ creates a concentration effect. For the strictly convex function $\phi(x) = e^{-x}$, Jensen's inequality yields:

$$\sum_i \tilde{g}_i \phi\left(\frac{g_i}{\tilde{g}_i \cdot S_g}\right) > \phi\left(\frac{1}{S_g}\sum_i g_i\right), \tag{16}$$

where $S_g = \sum_j g_j$. This convexity induces a **sharpening effect**:

- Small g_i (high similarity) $\to$ large α_i (high weight);
- Large g_i (low similarity) $\to$ small α_i (low weight).

The resulting distribution $\boldsymbol{\alpha}$ is more concentrated than $\tilde{\mathbf{g}}$, leading to:

$$H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{g}}). \tag{17}$$

Combining both stages through transitivity:

$$H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{g}}) \leq H(\tilde{\mathbf{c}}). \tag{18}$$

Therefore: $H(\boldsymbol{\alpha}) \leq H(\tilde{\mathbf{c}})$, completing our derivation.

Information-Theoretic Implications

The entropy reduction $\Delta H = H(\tilde{\mathbf{c}}) - H(\boldsymbol{\alpha}) \geq 0$ quantifies the **uncertainty reduction** achieved by our reweighting mechanism. This entropy reduction manifests as enhanced focus, where weights concentrate on tokens with extreme similarity values, while tokens with moderate similarity receive reduced attention. Consequently, critical tokens exhibiting high similarity to the special token are emphasized, leading to improved information preservation during the compression process. □

3.2.3 Tracer LLM

The Tracer LLM processes the compressed semantic token embeddings from the Inter LLM to capture temporal dependencies among students' learning interactions. To follow the sequential nature of KT, we utilize a transformer architecture with causal attention, ensuring that predictions at time step t only access information from previous time steps 1 to $t-1$.

$$\mathbf{E}_{<t} = [\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_{t-1}], \tag{19}$$

where $\mathbf{E}_{<t} \in \mathbb{R}^{(t-1)\times d_T}$ is the set of interaction token embeddings up to time step $t-1$. Formally, the Tracer LLM computes:

$$[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{t-1}] = \text{Tracer LLM}(\mathbf{E}_{<t}), \tag{20}$$

where $\mathbf{h}_{t-1}$ represents the student's knowledge state at time step $t-1$. These knowledge states capture the student's learning behaviors and evolving knowledge mastery based on their historical interaction patterns.

To illustrate the computational efficiency of our hierarchical architecture, consider a student's ID-based interaction sequence of length N, where each interaction

is converted into a textual description of length L. LLM4KT's computational cost consists of two components: (1) the Inter LLM processes each interaction independently, requiring $O(N \cdot L^2)$ operations, and (2) the Tracer LLM processes condensed N token embeddings with a complexity of $O(N^2)$. The total computational cost is: $O(NL^2 + N^2)$. Without the Inter LLM, processing this length of textual input would require $(NL)^2$ operations, similar to previous LLM-based KT methods. Additionally, the advantage of our architecture is that we do not need to adapt the dialogue-driven formats. For example, CLST converts a learning sequence of length N into N-turns dialogues, and inputs historical descriptions of lengths from $1 \times L$ to $(N-1) \times L$ to predict the response for corresponding upcoming questions iteratively. Therefore, processing this growing inputs requires computational costs of $L^2, (2L)^2, (3L)^2, ..., (NL)^2$ for all N-turn dialogues. The total computational cost thus becomes: $\sum_{i=1}^{N}(iL)^2 = L^2 \sum_{i=1}^{N} i^2 = O(N^3L^2)$, which represents a dramatic addition in computational complexity compared to our LLM4KT.

3.2.4 Functional Query/Recall LLM

As illustrated in Figure 2 b), the Query LLM utilizes a query prompt template to describe question q_t for the primary performance prediction task. This prompt includes question ID, concept ID, and timestep information, framed as a binary prediction problem. The task-related descriptions of q_t are then fed into the Query LLM. The Query LLM extracts the token embedding at the position of the special token "[QUERY]" and employs TRM$(\cdot)$ to refine the condensed token embeddings.

$$\mathbf{E}_t^{\text{weight}} = \text{TRM}(\mathbf{E}_t^{\text{query}}), \tag{21}$$

$$\mathbf{Q}_t = \text{Aligner}(\mathbf{E}_t^{\text{weight}}), \tag{22}$$

where $\mathbf{E}_t^{\text{query}}$ is the last hidden states of the Query LLM, $\mathbf{Q}_t \in \mathbb{R}^{1 \times d_T}$ denotes the semantic embedding of q_t.

The Recall LLM is designed to support the auxiliary question recall task. We design a recall prompt template that facilitates the Tracer LLM to understand the task. Like the Query LLM, the Recall LLM utilizes the specialized description of q_t as input.

$$\mathbf{E}_t^{\text{weight}} = \text{TRM}(\mathbf{E}_t^{\text{recall}}), \tag{23}$$

$$\mathbf{R}_t = \text{Aligner}(\mathbf{E}_t^{\text{weight}}), \tag{24}$$

where $\mathbf{E}_t^{\text{recall}}$ represents the last hidden states of Recall LLM, $\mathbf{R}_t \in \mathbb{R}^{1 \times d_T}$ denotes the semantic embedding of q_t for supporting the auxiliary task.

3.2.5 Feature Aligner

Inspired by adapter modules commonly used in transfer learning [31], we introduce a feature Aligner network Aligner$(\cdot)$ that ensure dimensional compatibility and semantic space alignment between different components with varying hidden sizes. Specifically, the Aligner network is implemented as a feed-forward neural network with a residual

connection:

$$\text{Aligner}(\mathbf{E}) = \text{FFN}(\mathbf{E}) + \text{Projection}(\mathbf{E}), \tag{25}$$

where $\mathbf{E} \in \mathbb{R}^{1\times d_1}$ denotes the input token embedding. The FFN$(\cdot)$ component is a two-layer feed-forward network with non-linear activation, while Projection$(\cdot)$ provides a linear transformation for dimensional matching::

$$\text{FFN}(\mathbf{E}) = \text{Dropout}(\text{GELU}(\mathbf{E}\mathbf{W}_1))\mathbf{W}_2, \tag{26}$$

$$\text{Projection}(\mathbf{E}) = \mathbf{E}\mathbf{W}_3, \tag{27}$$

where $\mathbf{W}_1, \mathbf{W}_3 \in \mathbb{R}^{d_I \times d_T}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_T \times d_T}$ are trainable parameter matrices. The residual connection preserves important information from the original embedding while enabling the aligner to learn effective transformations for different tasks.

3.3 Training Objectives

LLM4KT is fine-tuned with two objectives: the primary performance prediction task and the auxiliary question recall task. For the primary task, we use the hidden state $\mathbf{h}_{t-1}$ along with the query embedding $\mathbf{Q}_t$ generated from Query LLM to predict the student's response r_t to question q_t:

$$r_t = \sigma(f_{\text{pred}}([\mathbf{h}_{t-1}, \mathbf{Q}_t])), \tag{28}$$

where $[\cdot,\cdot]$ denotes the concatenation operation, $f_{\text{pred}}(\cdot)$ is a prediction head consisting of three fully connected layers with ReLU activations and dropout regularization, and $\sigma(\cdot)$ is the sigmoid function. The performance prediction objective $\mathcal{L}_{\text{pred}}$ uses binary cross-entropy loss to optimize the model's parameters.

$$\mathcal{L}_{\text{pred}} = -\frac{1}{|\mathcal{D}|}\sum_{q_t\in\mathcal{D}} [r_t \log(\hat{r}_t) + (1 - r_t)\log(1 - \hat{r}_t)], \tag{29}$$

where $\mathcal{D}$ is the set of all valid interactions (excluding padding), $\hat{r}_t \in \{0, 1\}$ is the ground truth response (1 for correct, 0 for incorrect), and $r_t \in [0, 1]$ is the predicted probability of a correct response.

For the auxiliary task, we use $\mathbf{h}_{t-1}$ and the embedding $\mathbf{R}_t$ generated from Recall LLM to predict the probability m_t whether the student has previously practised the knowledge concept c_t covered by q_t:

$$m_t = \sigma(f_{\text{rec}}([\mathbf{h}_{t-1}, \mathbf{R}_t])), \tag{30}$$

where $f_{\text{rec}}(\cdot)$ is another multi-layer perceptron with the same architecture as $f_{\text{pred}}(\cdot)$, but trained for the auxiliary task. The auxiliary objective $\mathcal{L}_{\text{recall}}$ also uses binary

cross-entropy loss to optimize the parameters:

$$\mathcal{L}_{\text{recall}} = -\frac{\sum_{q_t \in \mathcal{D}} [m_t \log(\hat{m}_t) + (1 - m_t) \log(1 - \hat{m}_t)]}{|\mathcal{D}|}, \tag{31}$$

where $\hat{m}_t \in \{0, 1\}$ is a binary indicator equal to 1 if the student has previously encountered a question affiliated to concept c_t before, and 0 otherwise. $m_t \in [0, 1]$ is the predicted probability that the student has encountered the concept before. The overall training objective is the sum of these two losses:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \cdot \mathcal{L}_{\text{recall}}, \tag{32}$$

where λ stands for the tradeoff parameters.

4 Experiments

In this section, we first introduce the basic experimental settings, and then conduct extensive experiments to assess the following research questions:

- **RQ1**: How effective and superior is the proposed LLM4KT framework in addressing the KT task?
- **RQ2**: Do the proposed key innovations contribute to enhancing the performance of our proposed LLM4KT model?
- **RQ3**: Does LLM4KT exhibit scalability with the model size of Inter LLM and Tracer LLM?
- **RQ4**: How robustness is our LLM4KT to different setting?

4.1 Settings

We evaluate LLM4KT on five widely-used benchmark datasets: Assist09, Algebra05, Statics, Ednet, and Spanish. We compare LLM4KT with 13 baseline models across two categories: (1) **Non-LLM KT models**: DKT [26], SAKT [25], AKT [7], ATKT [8], DTransformer [35], SimpleKT [18], FoLiBiKT [15], SparseKT [11], and DisKT [39]; (2) **LLM-based KT models**: FT Ext [24], LOKT [13], CLST [12], and LKT [14]. For our LLM4KT-4.6B, we use Qwen3-Embed-0.6B [34] as Inter LLM and Qwen3-4B [34] as Tracer LLM.

4.1.1 Dataset Descriptions

Following the data preprocessing approach in [39], we exclude students with fewer than five interactions and all interactions involving nameless concepts. For each dataset, if a student has more than 100 learning interactions, we break up their entire sequence into multiple shorter sequences.

Assist09: Originating from the ASSISTments online tutoring platform, this dataset captures mathematics learning interactions across 3,644 students tackling 17,727 questions spanning 123 distinct knowledge concepts. The dataset's popularity stems from its realistic representation of student learning patterns in STEM education.

Table 1 Statistics of the datasets used in experiments.

Datasets	#students	#questions	#concepts	#interactions
Assist09	3,644	17,727	123	281,890
Algebra05	571	173,113	112	607,014
Statics	333	1,223	N/A	189,297
Ednet	5,000	12,117	189	676,276
Spanish	182	409	221	578,726

Algebra05: This Algebra dataset presents an interesting contrast with its relatively small student cohort (571), yet massive question bank (173,113 questions) covering 112 algebraic concepts. The dataset's density makes it particularly valuable for studying learning patterns with extensive practice opportunities.

Statics: A specialized dataset from engineering education with 333 students and 1,223 questions. What sets Statics apart is its lack of explicit concept annotations, presenting unique challenges for traditional concept-based approaches while offering insights into domain-specific learning behaviors.

Ednet: Representing one of the largest KT datasets, Ednet encompasses 5,000 students across 12,117 questions and 189 concepts from a comprehensive online learning platform. Its scale and diversity make it particularly suitable for evaluating model robustness across varied educational contexts.

Spanish: This language learning dataset, while smaller in scale (182 students, 409 questions), offers valuable insights into KT for linguistic skills across 221 Spanish language concepts, providing a contrasting domain to STEM-focused datasets.

4.1.2 Baseline Models

Non-LLM KT Models:

- **DKT** [26]: The pioneering deep learning approach that introduced recurrent neural networks to capture temporal dependencies in student learning sequences, establishing the foundation for modern KT methods.
- **SAKT** [25]: builds upon the success of self-attention mechanisms, employing transformer architectures to model long-range dependencies that were challenging for earlier RNN-based approaches.
- **AKT** [7]: introduces attention mechanisms to dynamically weight historical interactions, allowing the model to focus on the most relevant past experiences when making predictions.
- **ATKT** [8]: enhances model robustness and generalization by jointly training with original and adversarial samples generated via perturbed interaction embeddings, using an efficient attention LSTM as the backbone network.
- **DTransformer** [35]: uses a contrastive learning-based training algorithm to enhance stability and accuracy in tracking knowledge states and predicting student performance.

- **SimpleKT** [18]: represents question embeddings as a combination of knowledge component embeddings and question-specific variations. It uses dot-product attention for student knowledge state estimation.
- **FoLiBiKT** [15]: introduces a linear bias to penalize question attention scores proportionally to the relative distance, effectively decoupling question correlations from forgetting behavior.
- **SparseKT** [11]: proposes soft-thresholding sparse attention and top-K sparse attention to help the DLKT models focus on the most influential interactions, reducing noise from irrelevant data.
- **DisKT** [39]: Introduces the concept of disentanglement to KT, separating mastery levels across different knowledge concepts for more interpretable and accurate modeling.

LLM-based KT Models:

- **FT Ext [24]:** uses LLMs without any fine-tuning on the specific task, simply by providing prompts related to student interactions and predicting performance based on these prompts.
- **LOKT** [13]: leverages textual option weights to distill long-context information into a compact sequence, enhancing knowledge tracing performance in a cost-effective and efficient manner.
- **CLST** [12]: utilizes a generative LLM as a knowledge tracer to predict student performance. It converts the KT task into a natural language processing task by expressing problem-solving data in natural language.
- **LKT** [14]: bridges traditional KT methods with modern language models by converting numerical sequences to text format and fine-tuning encoder-based PLMs like BERT and RoBERTa.

4.1.3 Implementation Details

For the entire dataset, the training ratio is 80%, and the test ratio is 20%. We adopt 10% of the training set for validation and an early stopping strategy, that is, if AUC does not improve within 5 epochs, the training is halted. For each dataset, if a student has more than 100 learning interactions, we break up their entire sequence into multiple shorter sequences. All models are optimized by AdamW [19] with the following settings: the learning rate is set to 0.001, and the weight decay is set to 0.01. All models are trained in PyTorch on a Linux server with eight GeForce A800 GPUs with 80GB of memory. To ensure reproducibility, we fix the random seed to 224 for all experiments and set PyTorch's deterministic mode. For efficiency, we use PyTorch's Automatic Mixed Precision [20] training with FP16 precision. The evaluation metrics include Area Under the ROC Curve (AUC), Accuracy (ACC), and Root Mean Square Error (RMSE).

For fair comparison, we use Qwen3-8B [34] as the backbone for CLST (larger than our LLM4KT-4.6B), and DeBERTa-v3 [9] as the backbone for LKT since it achieved the best performance reported in the paper. FT Ext and LOKT both use locally deployed Qwen3-32B [34] for testing. To optimize parameter efficiency and training stability for large language models, we employ LoRA [10] (Low-Rank Adaptation)

with rank $r = 32$, alpha parameter set to 1, and dropout rate of 0.1 for selective parameter updates. This configuration allows us to fine-tune the pre-trained language models while maintaining computational efficiency and preventing catastrophic forgetting. The LoRA adapters are applied to the attention layers of both Inter LLM and Tracer LLM components, enabling task-specific adaptation without modifying the full model parameters. For distributed training, we leverage data parallelism across multiple GPUs to handle the computational demands of large-scale language models. The training process supports both single-node and multi-node configurations, with gradient synchronization handled automatically through PyTorch's distributed data parallel framework. We implement gradient accumulation to effectively increase batch sizes when memory constraints limit the per-GPU batch size, particularly beneficial for fine-tuning larger language models.

4.2 Comparison Experiments (RQ1)

Table 2 presents the comprehensive performance comparison across all five datasets, from which we can observe several important findings: (1) Our proposed LLM4KT-4.6B achieves the best performance across all datasets, exhibiting excellent performance improvements. Specifically, LLM4KT-4.6B shows substantial improvements over the best baseline models, with absolute improvements ranging from +0.82% to +4.14% in AUC, +1.55% to +3.90% in ACC, and -0.82% to -3.67% in RMSE. These notable improvements validate the superiority of our novel hierarchical structure compared to the best baseline models. (2) Among LLM-based KT models, fine-tuning approaches (CLST, LKT) consistently outperform prompt-based methods (FT Ext, LOKT). For instance, on all five datasets, LKT achieves substantially higher performance compared to LOKT and FT Ext, demonstrating the necessity and effectiveness of task-specific adaptation through post-training strategy. (3) LLM-based methods (LKT, CLST) generally outperform most non-LLM approaches across datasets, despite having simpler architectural designs. For example, LKT achieves superior performance on multiple datasets compared to previous DLKT methods like FoLiBiKT and even sophisticated approaches like DisKT. Even when improvements are limited, these results demonstrate the powerful potential of LLMs for solving the KT task.

4.3 Ablation Experiments (RQ2)

4.4 Component Contribution Study

To investigate the contribution of our key innovations, including the auxiliary question recall task (QR) and token reweighting mechanism (TRM), we conduct additional experiments examining the impact of removing QR and TRM. From the experimental results shown in Table 3, we can observe several important findings: (1) Removing either component degrades performance, validating the effectiveness of both key innovations. Specifically, averaged across datasets, the AUC/ACC drops are approximately 1.54%/1.20% when both are removed, 1.13%/0.78% when only TRM is removed, and 0.65%/0.41% when only QR is removed. (2) The degradation is slightly larger when QR is disabled compared to when TRM is disabled on several datasets (e.g., Algebra05 and Assist09), indicating that explicit recall supervision contributes a bit more

Table 2 Performance comparison of different models on various datasets. The best performances for each metric are denoted in bold, and the second is underlined. For AUC and ACC, higher values are better (↑), while for RMSE, lower values are better (↓). The percentages (**% improve**) are absolute improvements to the best baseline model. "*" denotes the model without fine-tuning.

Datasets	**Assist09**			**Algebra05**			**Statics**			**Ednet**			**Spanish**		
Metrics	**AUC↑**	**ACC↑**	**RMSE↓**	**AUC↑**	**ACC↑**	**RMSE↓**	**AUC↑**	**ACC↑**	**RMSE↓**	**AUC↑**	**ACC↑**	**RMSE↓**	**AUC↑**	**ACC↑**	**RMSE↓**
non-LLM KT models															
DKT	0.7591	0.7166	0.4333	0.7780	0.7895	0.3854	0.7611	0.7643	0.4027	0.6589	0.6289	0.4771	0.8029	0.7433	0.4156
SAKT	0.7348	0.7000	0.4442	0.7493	0.7800	0.3942	0.7304	0.7560	0.4148	0.6499	0.6240	0.4809	0.7950	0.7417	0.4236
AKT	0.7705	0.7192	0.4349	0.7932	0.7938	0.3811	0.7999	0.7761	0.3889	0.7003	0.6592	0.4746	0.8391	0.7745	0.3968
ATKT	0.7543	0.7171	0.4348	0.7624	0.7882	0.3899	0.7421	0.7658	0.4194	0.6544	0.6243	0.4797	0.8047	0.7545	0.4186
DTransformer	0.7508	0.7042	0.4505	0.7694	0.7877	0.3906	0.7690	0.7732	0.3966	0.6978	0.6559	0.4799	0.8170	0.7513	0.4108
SimpleKT	0.7709	0.7209	0.4372	0.7874	0.7927	0.3842	0.7888	0.7723	0.3924	0.7048	0.6573	0.4730	0.8408	0.7734	0.3963
FoLiBiKT	0.7710	0.7165	0.4356	0.7923	0.7925	0.3818	0.7988	0.7794	0.3894	0.6995	0.6582	0.4756	0.8399	0.7735	0.3962
SparseKT	0.7670	0.7092	0.4396	0.7806	0.7856	0.3875	0.7887	0.7775	0.3909	0.7006	0.6557	0.4727	0.8395	0.7718	0.3959
DisKT	0.7923	0.7275	0.4294	0.8033	0.8009	0.3786	0.8086	0.7883	0.3823	0.7384	0.6863	0.4592	0.8524	0.7847	0.3872
LLM-based KT models															
FT Ext*	0.6454	0.6211	0.6155	0.6096	0.7131	0.5357	0.5990	0.7282	0.5214	0.6356	0.5966	0.4850	0.6426	0.6293	0.6089
LOKT*	0.6517	0.6730	0.5718	0.6246	0.7413	0.5086	0.6053	0.7226	0.5267	0.6528	0.6069	0.4807	0.6636	0.6680	0.5752
CLST	0.7676	0.7428	0.4216	0.7982	0.7908	0.3832	0.7984	0.7773	0.3891	0.7278	0.7017	0.4412	0.8386	0.7865	0.3928
LKT	0.8056	0.7547	0.4081	0.8155	0.8001	0.3725	0.8118	0.7882	0.3780	0.7351	0.6721	0.4562	0.8352	0.7877	0.3878
LLM4KT-4.6B	**0.8428**	**0.7816**	**0.3859**	**0.8517**	**0.8206**	**0.3560**	**0.8532**	**0.8039**	**0.3648**	**0.7423**	**0.7172**	**0.4331**	**0.8708**	**0.8267**	**0.3505**
% Improve	+3.72%	+2.69%	-2.22%	+3.67%	+1.97%	-1.65%	+4.14%	+1.56%	-1.32%	+0.82%	+1.55%	-0.82%	+1.84%	+3.90%	-3.67%

Table 3 Validation of different key components of LLM4KT. ✓ denotes the component is enabled, and ✗ denotes it is removed.

Variants		Assist09		Algebra05		Ednet		%Drop (avg)	
QR	TRM	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
✗	✗	0.8289	0.7690	0.8377	0.8130	0.7325	0.7095	-1.54%	-1.20%
✗	✓	0.8317	0.7734	0.8420	0.8154	0.7354	0.7124	-1.13%	-0.78%
✓	✗	0.8362	0.7774	0.8455	0.8175	0.7390	0.7149	-0.65%	-0.41%
✓	✓	**0.8428**	**0.7816**	**0.8517**	**0.8206**	**0.7423**	**0.7172**	-0.00%	-0.00%

to final accuracy. Disabling both components leads to the largest performance decline, demonstrating that QR and TRM are complementary rather than redundant.

4.4.1 Contribution Analysis of Different Layers Within LLM4KT

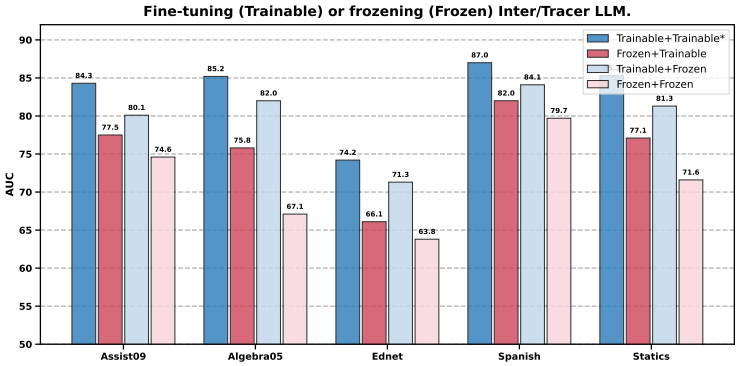


Fig. 3 Performance with different parameter optimization strategies for different layers within LLM4KT.

In this section, we further investigate the contribution of different layers within our two-layer hierarchical framework by selectively freezing or fine-tuning the Inter LLM and the Tracer LLM (excluding other components). Figure 3 presents the results across all five datasets for four different configurations: Trainable+Trainable, Trainable+Frozen, Frozen+Trainable, and Frozen+Frozen. From the experimental results, we can observe several important findings: (1) The Trainable+Trainable configuration yields the highest AUC across all five datasets, demonstrating the effectiveness of fine-tuning both layers. (2) Importantly, Trainable+Frozen configurations consistently outperform Frozen+Trainable configurations, suggesting that improving text understanding capacity for interaction descriptions is more critical than solely enhancing temporal modeling capacity. This may due to the fact that task-aligned semantic compression serves as a prerequisite for effective sequence modeling. (3) The Frozen+Frozen configuration performs the worst, underscoring the necessity of task-specific fine-tuning for both Inter LLM and Tracer LLM.

Table 4 Experiments with different sizes of the Inter LLM and Tracer LLM. "#P" denotes the parameter count.

Model	#P	Assist09			Algebra05		
		AUC	ACC	RMSE	AUC	ACC	RMSE
Inter LLM (Changed)+Tracer LLM(Qwen3-4B)							
Bert-base	110M	0.8317	0.7734	0.3951	0.8433	0.8140	0.3634
Bert-large	340M	0.8362	0.7774	0.3918	0.8481	0.8176	0.3580
Qwen3-Embed	0.6B	**0.8428**	**0.7816**	**0.3859**	**0.8517**	**0.8206**	**0.3560**
Qwen3-Embed	4B	0.8345	0.7751	0.3921	0.8479	0.8170	0.3601
Inter LLM(Qwen3-Embed-0.6B)+Tracer LLM(Changed)							
Qwen3	0.6B	0.8349	0.7762	0.3909	0.8470	0.8173	0.3599
Qwen3	1.7B	0.8387	0.7776	0.3881	0.8484	0.8201	0.3585

4.5 Scaling Up of Model Size (RQ3)

In this subsection, we study how scaling the Inter LLM and the Tracer LLM affects predictive performance. As shown in Table 4, when keeping the Tracer LLM fixed (Qwen3-4B), enlarging the Inter LLM from BERT-base to BERT-large, and further to Qwen3-Embed-0.6B leads to consistent performance improvements. However, when scaling up to Qwen3-Embed-4B, the expected gains don't materialize. Furthermore, when fixing the Inter LLM, increasing the Tracer LLM size consistently yields gains. This suggests that larger Tracer LLMs, with their enhanced temporal modeling capabilities, can better understand students' learning behaviors and achieve superior prediction performance based on the compressed semantic token embeddings.

To quantitatively assess the quality of compressed token embeddings generated by different Inter LLMs, we employ the Davies-Bouldin Index (DBI) [33], a widely-used metric for evaluating clustering quality. The DBI measures the ratio of within-cluster scatter to between-cluster separation, providing insights into how well-distributed and separable the embeddings are across different knowledge concepts. The DBI is calculated as follows:

$$\text{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right), \tag{33}$$

where k is the number of clusters (knowledge concepts), σ_i represents the average distance of all points in cluster i to their cluster centroid c_i, $d(c_i, c_j)$ is the distance between centroids of clusters i and j.

The DBI provides several key insights for our analysis: (1) When embeddings belonging to the same knowledge concept are tightly clustered (low σ_i) and different concepts are well-separated (high $d(c_i, c_j)$), the DBI decreases, signaling more effective semantic compression. (2) A lower DBI suggests that the Inter LLM has successfully learned to map similar learning interactions to nearby regions in the embedding space while keeping different knowledge concepts distinct, which is crucial for the subsequent Tracer LLM's temporal modeling capabilities. This aligns with our prompt analysis

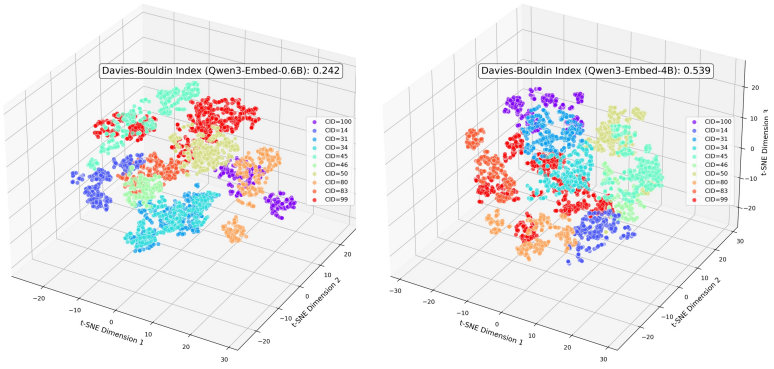


Fig. 4 Vector Space Visualization and DBI Analysis.

findings in Section 4.7.1, where we demonstrate that concept information is critical for model performance, with its removal causing the largest performance degradation (5.24% average AUC drop). Therefore, the ability to generate embeddings that effectively distinguish between different knowledge concept interactions is important to achieving optimal KT performance. (3) By comparing DBI values across different Inter LLM scales, we can objectively assess whether larger models actually produce more meaningful semantic representations, or if they suffer from overfitting or insufficient adaptation to the KT task.

The semantic compression process carried out by the Inter LLM is crucial for capturing the differentiation of student interactions. To explore why utilizing Qwen3-Embed-4B as Inter LLM degrades the performance despite its larger-scale parameters, we quantify the distribution quality of the condensed token embeddings generated by different Inter LLMs. As shown in Figure 4, a lower DBI [33] value indicates that the clusters are more distinct from each other, which suggests better separability of interactions affiliated to different knowledge concepts. The results demonstrate that the spatial distribution of students' interaction descriptions compressed by Qwen3-Embed-4B is not as effective as those compressed by Qwen3-Embed-0.6B, potentially creating difficulties for the subsequent Tracer LLM when accurately predicting students' future performance. Based on these experimental observations, we conjecture that this performance degradation stems from insufficient parameter updating of Qwen3-Embed-4B on the current dataset size, leading to suboptimal semantic compression quality.

4.6 Robustness Study (RQ4)

4.7 Sensitivity Analysis of Hyper-parameter

In order to validate the effectiveness of our proposed auxiliary task and understand its impact on the primary performance prediction task, we investigate how the hyper-parameter λ balances the above two objectives. We vary $\lambda \in \{0.1, 0.5, 2, 8, 32\}$ to examine the trade-off between the two tasks. As shown in Figure 5, when $\lambda < 8$, the AUC of performance prediction remains virtually flat (fluctuation $< 1\%$), demonstrating that the model is robust to parameter settings and that the auxiliary task

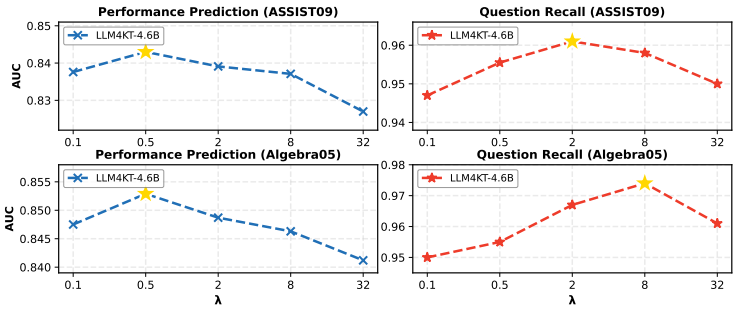


Fig. 5 Ablation Studies on the Impact of Different λ Values on Performance Prediction and Question Recall Tasks.

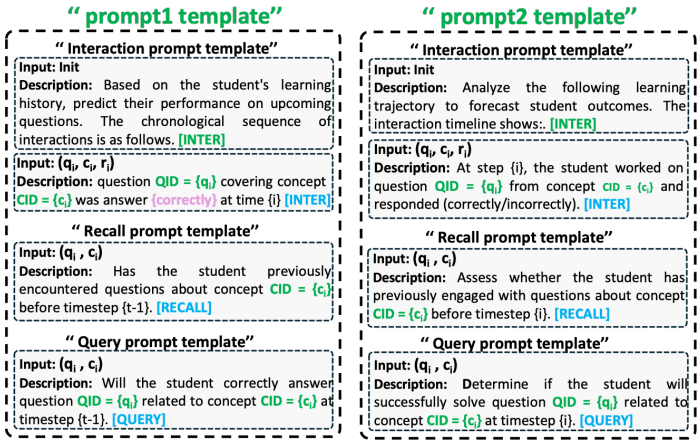


Fig. 6 The template of prompt1 and prompt2.

effectively promotes the primary task performance without requiring careful hyper-parameter tuning. Concurrently, the AUC results of the question recall task remain in a high band of $[0.94, 0.98]$, which exhibits a growing trend with the λ increasing. As λ is enlarged beyond 8, the performance of both tasks drops, which suggests that an excessively large λ biases optimization direction, leading to slightly overfit to the simpler auxiliary task while progressively overlooking the primary task.

4.7.1 Robustness Analysis of Prompts

To evaluate the robustness of our model to different prompt formulations, we design four prompt variants that test different aspects of prompt sensitivity: two alternative formulations (Prompt1 and Prompt2) that maintain all semantic components shown in Figure 6, and two ablated versions that remove critical information elements within the original prompt used by LLM4KT-4.6B (wo/time removes all temporal sequence information including time step references, while wo/concept eliminates concept ID information while preserving question-level details).

Table 5 Effect of Different Prompt Designs on Model Performance of LLM4KT-4.6B (Ours).

Variants	Assist09		Algebra05		Ednet		%Drop (avg)	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
Ours	**0.8428**	**0.7816**	0.8517	0.8206	**0.7423**	**0.7172**	-0.0%	-0.0%
Prompt1	0.8420	0.7795	**0.8520**	**0.8212**	0.7411	0.7155	-0.057%	-0.11%
Prompt2	0.8415	0.7804	0.8511	0.8199	0.7418	0.7160	-0.08%	-0.1%
wo/time	0.8233	0.7710	0.8318	0.8087	0.7320	0.6986	-1.65%	-1.37%
wo/concept	0.7919	0.7552	0.7890	0.7853	0.6987	0.6718	-5.24%	-3.57%

As shown in Table 5, our LLM4KT-4.6B is robust to variations in different prompts, as the performance remains stable between the original prompt (LLM4KT-4.6B) and its two alternatives (Prompt1, Prompt2). However, the model's performance is highly sensitive to the valid content of the prompt text. Removing key semantic components like temporal information (wo/time) or concept information (wo/concept) leads to a significant drop in performance across all metrics. The absence of temporal information (wo/time) results in a more noticeable decrease in AUC and ACC, averaging around 1.65% and 1.37% respectively across the datasets. This indicates that temporal information, which reflects the sequence of student interactions with knowledge points, is crucial for accurate knowledge tracing. The most significant drop in performance is observed when concept information is omitted (wo/concept), with an average decrease of 5.24% in AUC and 3.57% in ACC. This is because the number of questions greatly exceeds the number of concepts, leading to a sparse representation when only questions are used. This sparsity hinders the model's ability to effectively capture and utilize the underlying knowledge structure, which is essential for tracing student knowledge progression accurately. This confirms that while the specific phrasing is not critical, the presence of these essential informational components in the prompt is crucial for our model to effectively trace student knowledge.

Building upon the results of the impact of different prompt designs on model performance shown in Table 5, we conducted a further investigation to understand the intrinsic contributions of individual tokens within the prompts. This analysis was motivated by the need to delve deeper into the mechanisms by which certain prompts influence the model's predictive capabilities. Figure 7 presents a detailed examination of the cosine similarity between special tokens and other tokens within the sequence for three distinct prompt templates within Inter/Query/Recall LLMs. The red dots mark the position of the special token, and the dashed red line indicates the mean cosine similarity across all positions. The plots reveal that tokens related to 'question', 'concept', 'time', and the indicators of 'correct' or 'incorrect' responses generally exhibit higher similarity values, underscoring their significant role in the model's predictive tasks. The findings from this analysis not only validate the importance of including temporal and concept information in the prompts but also highlight the potential for enhancing model performance through the strategic reweighting of these critical tokens.



Fig. 7 Analysis of Token Importance via Cosine Similarity.

4.7.2 Few-Shot/Length Robustness Study

To further evaluate the robustness and practical applicability of our proposed LLM4KT, we conduct two complementary experiments: (1) a few-shot learning analysis to examine predictive performance under data scarcity scenarios, and (2) a sequence-length robustness evaluation to assess how well the model performs in varying length intervals. For the few-shot learning analysis, we train LLM4KT-4.6B and other models using varying proportions of the ASSIST09 and Algebra05 datasets. As

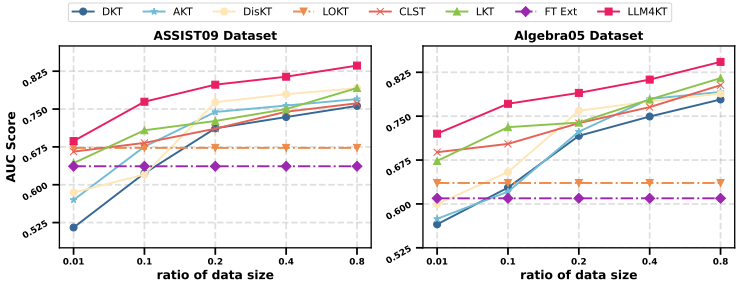


Fig. 8 Comparison across varying training-data ratio.

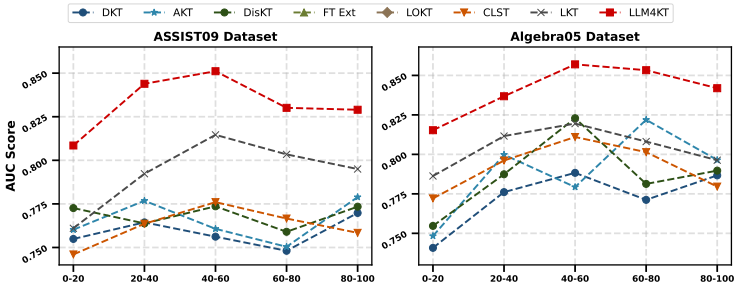


Fig. 9 Comparison across varying length intervals.

shown in Figure 8, the results reveal three key insights: (1) Even with only 1% of the training data, our LLM4KT significantly outperforms prompt-based LOKT and FT Ext (implemented with Qwen3-32B), demonstrating that task-specific fine-tuning is essential for achieving success in KT. This highlights the limitations of relying solely on in-context learning for understanding complex learning behaviors. (2) Under the same fine-tuning conditions, LLM4KT consistently outperforms LKT and CLST across all data ratios. This suggests that our hierarchical architecture, which decomposes semantic understanding and temporal dependency modeling, enables superior adaptation to low-resource training settings. (3) LLM4KT exhibits superior data efficiency, achieving the highest performance gains as the size of the training data increases. This indicates that our framework excels in low-resource settings compared to all competing approaches.

To explore how robustness our LLM4KT perform across different interaction length intervals (0–20, 20–40, 40–60, 60–80, and 80–100), particularly in long-term intervals that are known to challenge the modeling capacity of LLM-based KT approaches. As illustrated in Figure 9, we observe: (1) Most LLM-based models exhibit a decline in performance as sequence length increases. This is largely attributed to the growing length of textual inputs, which increases the difficulty of attention allocation and information extraction, thereby impairing the understanding of students' learning behaviors. (2) In contrast, our LLM4KT maintains strong performance in varying intervals. This robustness can be attributed to our hierarchical design that effectively decomposes complex, lengthy interaction histories into compressed semantic token embeddings.

4.7.3 Efficiency Study

Beyond predictive accuracy, we further investigate the computational efficiency of LLM4KT to empirically validate the theoretical complexity analysis in Section 3.2.3. Specifically, we measure the *per-sequence training time* and the *peak GPU memory* as the interaction sequence length N grows from 10 to 50. We compare two variants of our framework, LLM4KT-1.2B and LLM4KT-4.6B, against the representative dialogue-driven LLM-based baseline CLST, which adapts N-turn dialogue formulation. All experiments are conducted on a single NVIDIA A800-80G GPU, mixed-precision training, and identical optimizer settings, so that the reported numbers reflect the intrinsic cost of each model rather than implementation tricks. For every value of

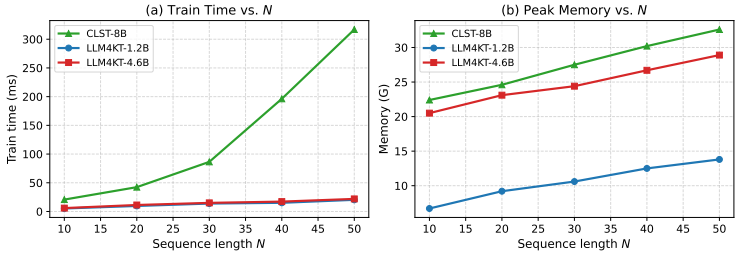


Fig. 10 Per-sequence training time and peak GPU memory of LLM4KT-1.2B, LLM4KT-4.6B and CLST under batch size 1 as the interaction sequence length N grows from 10 to 50.

$N \in \{10, 20, \ldots, 50\}$ we record the per-sequence training time on ASSIST09 together with the peak memory consumption. Figure 10 reports the results, from which we draw the following observations. (1) Per-sequence training time grows mildly with N for LLM4KT. Although both LLM4KT-1.2B and LLM4KT-4.6B exhibit an upward trend as the sequence becomes longer, the slope is rather flat. This is because the Tracer LLM operates on the condensed semantic token embeddings produced by the Inter LLM, so its per-step computation does not scale with the original textual length L. In contrast, CLST repeatedly feeds growing textual histories into the backbone for each of the N dialogue turns, leading to a markedly steeper increase in training time as N grows—by N=50 it is already an order of magnitude slower than our models. (2) Peak memory increases with N but stays favorable for LLM4KT. Larger sequences inevitably require storing more activations and key/value caches, which is reflected by the monotonically growing memory curves of all three models. Nevertheless, both LLM4KT-1.2B and LLM4KT-4.6B keep their peak memory consistently below CLST across the entire range of N, while delivering substantially better predictive accuracy as reported in Table 2. Together, these observations confirm that, by decoupling semantic compression from temporal modeling, LLM4KT achieves a markedly more graceful efficiency–accuracy trade-off than directly feeding raw textual histories into a single LLM backbone.

5 Conclusion

In this paper, we present LLM4KT, a novel hierarchical framework that separates semantic information processing from temporal dependency modeling for KT. LLM4KT employs an Inter LLM for compressing interactions' text description and a Tracer LLM for tracking students' knowledge states. The token reweighting mechanism ensures critical information preservation during compression, while the auxiliary question recall task enhances predictive accuracy by explicitly guiding Tracer LLM to memorize students' prior exposure to knowledge concepts. This work not only establishes new state-of-the-art performance but also provides valuable insights into the effective integration of LLMs for KT.

Acknowledgements. This work was supported by the Outstanding Innovative Talents Cultivation Funded Programs 2025 of Renmin University of China. We thank all

contributing authors for their valuable comments and constructive suggestions, which significantly improved the quality of this manuscript.

Declarations

- Funding: Supported by the Outstanding Innovative Talents Cultivation Funded Programs 2025 of Renmin University of China.
- Conflict of interest/Competing interests: The authors declare no competing interests.
- Ethics approval and consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Code availability: The code is available at supplementary file.
- Author contribution: Shuaishuai Zu conceptualized the study, designed the methodology, performed the experiments, and wrote the original draft. Biao Qin supervised the research, reviewed and edited the manuscript, and provided funding acquisition. Jihao Zhao assisted with data curation, implemented the experimental framework, and conducted performance evaluation. Shuhao Guan provided critical feedback on the manuscript, contributed to result interpretation, and helped with manuscript revision.

References

[1] Abdelrahman G, Wang Q (2019) Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 175–184

[2] Abdelrahman G, Wang Q, Nunes B (2023) Knowledge tracing: A survey. ACM Computing Surveys 55(11):1–37

[3] Chang Y, Wang X, Wang J, et al (2024) A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology 15(3):1–45

[4] Cover TM (1999) Elements of information theory. John Wiley & Sons

[5] Fu L, Guan H, Du K, et al (2024) Sinkt: A structure-aware inductive knowledge tracing model with large language model. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp 632–642

[6] Ge T, Hu J, Wang L, et al (2023) In-context autoencoder for context compression in a large language model. arXiv preprint arXiv:230706945

[7] Ghosh A, Heffernan N, Lan AS (2020) Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2330–2339

[8] Guo X, Huang Z, Gao J, et al (2021) Enhancing knowledge tracing via adversarial training. In: Proceedings of the 29th ACM International Conference on Multimedia, pp 367–375

[9] He P, Liu X, Gao J, et al (2020) Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:200603654

[10] Hu EJ, Shen Y, Wallis P, et al (2022) Lora: Low-rank adaptation of large language models. ICLR 1(2):3

[11] Huang S, Liu Z, Zhao X, et al (2023) Towards robust knowledge tracing models via k-sparse attention. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 2441–2445

[12] Jung H, Yoo J, Yoon Y, et al (2024) Clst: Cold-start mitigation in knowledge tracing by aligning a generative language model as a students' knowledge tracer. arXiv preprint arXiv:240610296

[13] Kim J, Chu S, Wong B, et al (2024) Not all options are created equal: Textual option weighting for token-efficient llm-based knowledge tracing. arXiv preprint arXiv:241012872

[14] Lee U, Bae J, Kim D, et al (2024) Language model can do knowledge tracing: Simple but effective method to integrate language model and knowledge tracing task. arXiv preprint arXiv:240602893

[15] Li X, Bai Y, Guo T, et al (2024) Enhancing length generalization for attention based knowledge tracing models with linear biases. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), pp 5918–5926

[16] Li Y, Dong B, Lin C, et al (2023) Compressing context to enhance inference efficiency of large language models. arXiv preprint arXiv:231006201

[17] Liu J, Zhu D, Bai Z, et al (2025) A comprehensive survey on long context language modeling. arXiv preprint arXiv:250317407

[18] Liu Z, Liu Q, Chen J, et al (2023) simplekt: a simple but tough-to-beat baseline for knowledge tracing. arXiv preprint arXiv:230206881

[19] Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:171105101

[20] Micikevicius P, Narang S, Alben J, et al (2017) Mixed precision training. arXiv preprint arXiv:171003740

[21] Nagatani K, Zhang Q, Sato M, et al (2019) Augmenting knowledge tracing by considering forgetting behavior. In: The world wide web conference, pp 3101–3107

[22] Nakagawa H, Iwasawa Y, Matsuo Y (2019) Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp 156–163

[23] Neelakantan A, Xu T, Puri R, et al (2022) Text and code embeddings by contrastive pre-training. arXiv preprint arXiv:220110005

[24] Neshaei SP, Davis RL, Hazimeh A, et al (2024) Towards modeling learner performance with large language models. arXiv preprint arXiv:240314661

[25] Pandey S, Karypis G (2019) A self-attentive model for knowledge tracing. arXiv preprint arXiv:190706837

[26] Piech C, Bassen J, Huang J, et al (2015) Deep knowledge tracing. Advances in neural information processing systems 28

[27] Pu X, He T, Wan X (2024) Style-compress: An llm-based prompt compression framework considering task-specific styles. arXiv preprint arXiv:241014042

[28] Shen S, Liu Q, Huang Z, et al (2024) A survey of knowledge tracing: Models, variants, and applications. IEEE Transactions on Learning Technologies

[29] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

[30] Wang C, Ma W, Zhang M, et al (2021) Temporal cross-effects in knowledge tracing. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp 517–525

[31] Wang R, Tang D, Duan N, et al (2020) K-adapter: Infusing knowledge into pre-trained models with adapters. arXiv preprint arXiv:200201808

[32] Wang X, Chen Z, Xie Z, et al (2024) In-context former: Lightning-fast compressing context for large language model. arXiv preprint arXiv:240613618

[33] Xiao J, Lu J, Li X (2017) Davies bouldin index based hierarchical initialization k-means. Intelligent Data Analysis 21(6):1327–1338

[34] Yang A, Li A, Yang B, et al (2025) Qwen3 technical report. arXiv preprint arXiv:250509388

[35] Yin Y, Dai L, Huang Z, et al (2023) Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023, pp 855–864

[36] Yu Y, Si X, Hu C, et al (2019) A review of recurrent neural networks: Lstm cells and network architectures. Neural computation 31(7):1235–1270

[37] Zhang J, Shi X, King I, et al (2017) Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web, pp 765–774

[38] Zhao WX, Zhou K, Li J, et al (2023) A survey of large language models. arXiv preprint arXiv:230318223 1(2)

[39] Zhou Y, Lv Z, Zhang S, et al (2025) Disentangled knowledge tracing for alleviating cognitive bias. arXiv preprint arXiv:250302539

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Code.zip