LLM-KT: Enhancing Large Language Models with Knowledge Tracing via Multi-Level Plug-and-Play Alignment

Ziwei Wang[1], Jie Zhou[1]*, Qin Chen[1], Bo Jiang[2], Qingchun Bai[3]*, Liang Dou[1], Liang He[1],
[1]School of Computer Science and Technology, East China Normal University
[2]Shanghai Institute of Artificial Intelligence for Education, East China Normal University
[3]Shanghai Open University

Abstract

Knowledge Tracing (KT) is a pivotal task in personalized education, aiming to predict students' future performance based on their historical interactions. While prior work has focused on learning behavioral sequences using question IDs or surface-level textual features, these methods often fail to capture complex behavioral patterns due to a lack of deep reasoning capabilities and world knowledge. To address this, we propose **LLM-KT**, a novel framework that integrates the reasoning power of Large Language Models (LLMs) with the sequential modeling strengths of traditional KT methods via multi-level plug-and-play alignment. Specifically, for task-level alignment, we design a plug-and-play instruction to leverage the rich knowledge and reasoning capacity of LLMs for the KT objective. For modality-level alignment, we introduce two mechanisms to integrate representations learned by traditional methods: (1) a Semantic History Projector that flexibly inserts compressed context embeddings into LLMs using question- and concept-specific tokens to capture long-term history; and (2) a Behavioral Dynamics Projector that enhances LLMs with sequential interaction patterns via a sequence adapter. Extensive experiments on four standard datasets demonstrate that LLM-KT achieves state-of-the-art performance, significantly outperforming over 20 competitive baselines.

1 Introduction

Knowledge tracing (Abdelrahman et al., 2023; Zanellati et al., 2024; Shen et al., 2024) aims to infer students' performance based on their historical question-answer records for personalized education. This technique can help teachers and education systems understand the knowledge status of students, such as their skills and forgetting behavior. By doing so, it provides more accurate teaching plans and resources. Effectively solving the knowledge tracing problem can significantly enhance the efficiency of computer-aided education.

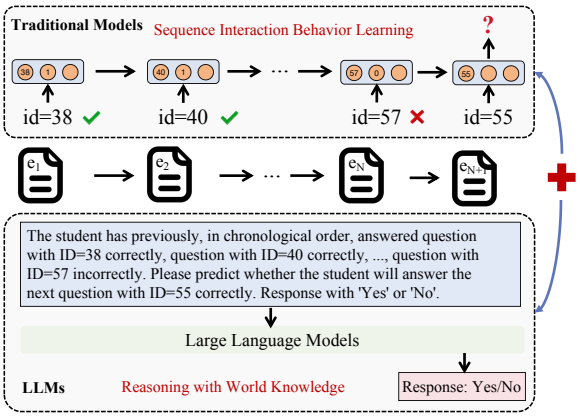

Figure 1: The advantages of traditional models and LLMs for KT. Traditional models are good at learning the sequence of interaction behavior, while LLMs are good at reasoning with rich world knowledge.

Traditional deep learning-based models mainly focus on modeling the interaction between questions using IDs (e.g., Question IDs or Concept IDs) to learn the sequence behavior information (See Figure 1). Sequence learning models (e.g., LSTM (Hochreiter, 1997) and Transformer (Vaswani, 2017)) are utilized to capture the representation of problem-solving records (Piech et al., 2015; Pandey and Karypis, 2019). Nagatani et al. (2019); Wang et al. (2021) integrate the time factor into the model to further learn the sequence information. Additionally, to learn the relationships among questions and concepts, graph neural networks are adopted (Song et al., 2022; Liu et al., 2021). These models enhance the interaction representations of an ID sequence using extra knowledge, such as time features and graph structure. However, the questions' textual information that contains rich seman-

[1]Corresponding authors, jzhou@cs.ecnu.edu.cn, qc_bai@foxmail.com

Findings of the Association for Computational Linguistics: ACL 2026, pages 35775–35792
July 2-7, 2026 ©2026 Association for Computational Linguistics

tic knowledge is not well explored.

Some studies have incorporated pre-trained language models (PLMs, such as BERT (Devlin et al., 2019)) into knowledge tracing to model the textual information of the question (Tan et al., 2022; Tong et al., 2020). For instance, BiDKT (Tan et al., 2022) adapts BERT to trace knowledge by predicting the correctness of randomly masked responses within sequences. MLFBK (Li et al., 2023) and LBKT (Li et al., 2024) leverage BERT to mine complex data relations. These methods use the question representations obtained from PLMs to enhance the traditional sequence models. Though PLMs are good at natural language understanding, they cannot effectively mine the logic and reasons behind the sequence of questions due to their limited reasoning and world knowledge acquisition abilities.

Recently, large language models (LLMs) like LLaMA (Touvron et al., 2023) have achieved great success in various natural language processing tasks due to their abilities in generation, instruction following, and reasoning. For knowledge tracing (KT), relevant research has explored two main application paradigms of LLMs: some studies focus on leveraging LLMs as a feature extractor to boost the performance of existing KT models (Liang et al., 2023; Ni et al., 2024; Xia et al., 2023), while others abandon traditional KT frameworks and adopt LLMs as standalone KT models that directly predict student status (Lee et al., 2024a; Li et al., 2025; Neshaei et al., 2024)[1].

Building on these explorations, we aim to integrate LLMs into knowledge tracing, motivated by two core advantages. **First**, LLMs possess *semantic reasoning* capabilities absent in ID-based methods: given a question text, an LLM can infer prerequisite concepts and judge relative difficulty, whereas conventional models only see opaque integer IDs. **Second**, the broad world knowledge acquired during pre-training endows LLMs with superior *generalizability*, making an LLM-enhanced framework inherently more portable to new domains and question sets than dataset-specific ID embeddings trained from scratch.

However, there are two primary challenges to applying LLMs for knowledge tracing. First (**C1**), LLMs struggle to capture sequential interaction behaviors from a series of IDs, which reflect students' knowledge states. Large language models interpret question IDs merely as numbers and fail to comprehend user behavior. This often results in splitting IDs into multiple tokens, causing a loss of semantic information associated with those IDs. Experiments indicate that fine-tuning LLaMA can enhance performance, but the gains are limited (refer to Table 1). Second (**C2**), LLMs have difficulty accurately capturing the long textual context of comprehensive problem-solving records. The textual content of questions and concepts is crucial for understanding user behavior. However, historical records may include over 200 questions, with each question averaging around 77 tokens. Experimental findings also suggest that the existing strong LLMs like LLaMA and GPT-4o cannot effectively learn students' states from this textual context.

To effectively synergize the sequential modeling capabilities of traditional methods with the reasoning power of LLMs, we propose a novel framework named **LLM-KT**. Specifically, we design a Multi-Level Plug-and-Play Alignment strategy to adapt LLMs for knowledge tracing. First, for task-level alignment, we construct instructions equipped with question- and concept-specific tokens to seamlessly incorporate heterogeneous data (e.g., texts and IDs). Addressing the challenge of modeling interaction behaviors (**C1**), we introduce a Behavioral Dynamics Projector module. This module projects the sequential interaction embeddings learned by traditional KT models into the LLM's semantic space, thereby enhancing the LLM's ability to interpret ID-based sequences. Addressing the challenge of long context (**C2**), we propose a Semantic History Projector module. Instead of feeding raw text directly, we compress the extended history into compact embeddings via a context adapter, allowing the LLM to efficiently capture semantic information from long-term records. Extensive experiments on four benchmark datasets demonstrate that **LLM-KT** consistently outperforms strong baselines. Furthermore, ablation studies validate the efficacy of our proposed components.

The main contributions of this paper are summarized as follows:

- We propose **LLM-KT**, a framework that aligns LLMs with knowledge tracing tasks via Multi-level Plug-and-Play Alignment. We introduce a flexible instruction tuning mechanism using specific tokens to bridge the gap between general reasoning and educational contexts.
- We design two distinct plug-in modules to handle

[1]More related studies are reviewed in Section A in the Appendix.

specific data modalities: a *Behavioral Dynamics Projector* that integrates interaction patterns from traditional sequence models, and a *Semantic History Projector* that efficiently captures long-term textual history via compressed embeddings.

- We conduct comprehensive experiments on four public benchmarks. The results show that **LLM-KT** achieves new SOTA performance compared to competitive baselines, demonstrating the superiority of our hybrid approach.

2 Our Proposed Method

In this section, we present **LLM-KT**, a novel framework specifically designed to adapt Large Language Models for knowledge tracing tasks (see Figure 2). To align LLMs with KT at the task level, we devise a plug-and-play instruction mechanism equipped with specialized tokens, which facilitates the flexible integration of heterogeneous modalities (e.g., long textual contexts and ID sequences). This design effectively synergizes the rich world knowledge and reasoning capabilities of LLMs with the sequential modeling strengths of traditional KT models. At the modality level, we introduce two core components: first, a Semantic History Projector that encodes long-term interaction history via a context adapter to align compressed representations with the LLM; and second, a Behavioral Dynamics Projector that enhances the LLM by injecting sequential interaction patterns learned from traditional models.

Knowledge tracing involves predicting whether a student will answer a new question correctly based on their historical question-answer records. Formally, given a student's exercise history as $H = (e_1, e_2, ..., e_i, ..., e_N)$, where N is the number of historical exercises. Here, $e_i = (q_i, a_i)$, where q_i represents the information of the i-th question the student answered and a_i indicates the student's response to this question ($a_i = 1$ means the student answered correctly, and $a_i = 0$ means the student answered incorrectly). The goal of the task is to predict the value of a_{N+1} (also defined as y) when the student answers q_{N+1}.

2.1 Plug-and-Play Instruction

To effectively bridge the gap between the discrete nature of ID-based interaction records and the continuous semantic space of Large Language Models, we design a Plug-and-Play Instruction mechanism. Traditional prompting methods that treat IDs as plain text often fail to preserve the underlying behavioral patterns, while full-text inputs suffer from excessive length. To address this, our approach adopts a *soft-prompting* paradigm: we construct a natural language template with specific placeholder tokens and physically substitute their input embeddings with the dense representations learned by our projectors. This strategy allows heterogeneous modalities—semantic context and behavioral dynamics—to be seamlessly integrated into the LLM's reasoning chain without compromising token limitations.

The data is organized into (input, output) pairs as illustrated below. The input prompts the model to analyze the student's learning trajectory, while the output is the predicted correctness label:

> **Input x.**
> The student has previously, in chronological order, answered {question QID=38 [QuesEmbed$_{38}$] involving concept CID=219 [ConcEmbed$_{219}$] correctly, ..., question QID=57 [QuesEmbed$_{57}$] involving concept CID=204 [ConcEmbed$_{204}$] incorrectly}$_{\text{HistoryRecord}}$.
> Please predict whether the student will answer the next {question QID=55 [QuesEmbed$_{55}$] involving concept CID=245 [ConcEmbed$_{245}$] correctly}$_{\text{TargetQues}}$. Response with 'Yes' or 'No'.
> Response:
> **Output y.**
> Yes/No

In this template, the placeholder tokens (e.g., [QuesEmbed]) serve as the interface for multimodal fusion. They correspond to the unified embeddings $\mathbf{e}_k$ for attribute $k \in \{q, c\}$, which are derived by synthesizing the semantic features from the Semantic History Projector (f_{cont}) and the interaction patterns from the Behavioral Dynamics Projector (f_{seq}). Formally, we employ an aggregation function $\mathcal{G}$ (e.g., gated fusion) to integrate these complementary representations:

$$\mathbf{e}_k = \mathcal{G}(\mathbf{h}_k^{\text{ctx}}, \mathbf{h}_k^{\text{seq}}), \quad \forall k \in \{q, c\} \tag{1}$$

By replacing standard vocabulary embeddings with these fused vectors $\mathbf{e}_k$, we ensure that the LLM perceives both the "meaning" of the question and the "history" of the interaction simultaneously.

Training Objective. We formulate knowledge tracing as a conditional text generation task. The model $\mathcal{M}_\theta$ accepts the prompt x and maximizes the likelihood of the correct label $\hat{y} \in \{\text{“Yes”}, \text{“No”}\}$.

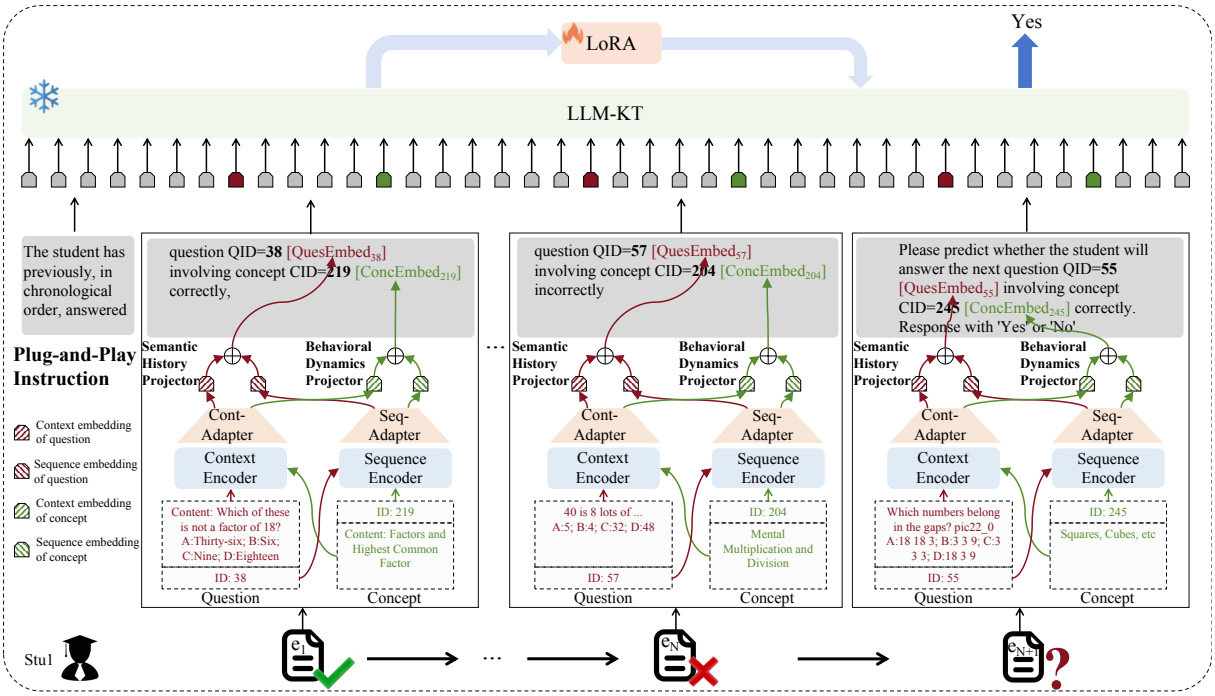

Figure 2: Overview of the **LLM-KT** framework. We propose a Multi-level Plug-and-Play Alignment strategy to synergize the reasoning power of LLMs with traditional sequence models. Specifically, the Semantic History Projector encodes the long-term context of student problem-solving records, while the Behavioral Dynamics Projector aligns the sequential interaction patterns learned by traditional models into the LLM's semantic space.

Since the injected embeddings $\mathbf{e}_k$ lie in a learned manifold distinct from the LLM's original space, we employ Low-Rank Adaptation (LoRA) (J. et al., 2021) to facilitate alignment. By optimizing the cross-entropy loss on the low-rank matrices while freezing the pre-trained weights, we enable the LLM to adapt to these new "knowledge-aware" tokens, achieving a balance between task-specific learning and general reasoning preservation.

During the inference phase, we map the class labels to a specific candidate token set $\mathcal{C} = \{v_{\text{yes}}, v_{\text{no}}\}$. The probability of the positive class is obtained by normalizing the model's likelihood over this candidate set:

$$P(\hat{y} = 1|x) = \frac{P_\theta(v_{\text{yes}}|x)}{P_\theta(v_{\text{yes}}|x) + P_\theta(v_{\text{no}}|x)} \quad (2)$$

where $P_\theta(v|x)$ denotes the probability assigned by the LLM to token v given context x.

2.2 Semantic History Projector

Processing the exhaustive history of student interactions presents a significant challenge for LLMs, as the combination of long interaction sequences and verbose question texts often exceeds the effective context window. Consequently, directly modeling raw textual history is computationally prohibitive and prone to information loss. To mitigate this, we introduce the Semantic History Projector, denoted as f_{cont}, to succinctly model the semantic information of questions and concepts. Specifically, this module comprises a *Context Encoder* to extract semantic features and a *Context Adapter* to align these representations with the LLM's latent space.

2.2.1 Context Encoder

We employ a pre-trained language model (PLM) as the backbone of our Context Encoder, denoted as $\mathcal{E}_{\text{ctx}}(\cdot)$, to transform raw textual inputs into dense semantic vectors. Various architectures, such as LLaMA2 (Touvron et al., 2023), BERT (Devlin et al., 2019), or all-mpnet-base-v2 (Song et al., 2020), can serve as the foundation.

Let T_k represent the textual content for a specific attribute $k \in \{q, c\}$, corresponding to questions and concepts, respectively. The encoder extracts the semantic embeddings $\mathbf{r}_k^{\text{ctx}} \in \mathbb{R}^{d^t}$ as follows:

$$\mathbf{r}_k^{\text{ctx}} = \mathcal{E}_{\text{ctx}}(T_k), \quad \forall k \in \{q, c\} \quad (3)$$

where d^t is the hidden dimension of the PLM. These embeddings effectively encapsulate the semantic nuances required to discern the complex relationships between educational content.

2.2.2 Context Adapter

Since the feature space of the PLM is distinct from that of the target LLM, direct injection leads to semantic misalignment. To bridge this gap, we introduce a Context Adapter, parameterized as a projection function $\mathcal{P}_\phi(\cdot)$. We implement this adapter using a Multi-Layer Perceptron (MLP) acting as a projection head. It maps the frozen text embeddings into the LLM's continuous embedding space:

$$\mathbf{h}_k^{\text{ctx}} = \mathcal{P}_\phi(\mathbf{r}_k^{\text{ctx}}), \quad \forall k \in \{q, c\} \tag{4}$$

where $\mathbf{h}_k^{\text{ctx}} \in \mathbb{R}^{d^e}$ denotes the aligned representations (with d^e being the LLM's embedding dimension). This transformation ensures that the rich semantic information captured by the encoder is compatible with the LLM's reasoning process.

2.3 Behavioral Dynamics Projector

While Large Language Models exhibit exceptional proficiency in natural language understanding, they often struggle to intrinsically capture the complex temporal dynamics embedded within abstract ID sequences. To address this limitation, we introduce the Behavioral Dynamics Projector (f_{seq}). This module is designed to encode interaction patterns as a distinct modality and project them into the LLM's latent space. By doing so, we effectively synergize the semantic reasoning capabilities of LLMs with the sequential modeling strengths of traditional KT methods, ensuring that both semantic context and behavioral history are utilized simultaneously.

2.3.1 Sequence Encoder

We treat the sequence of interaction IDs as a critical modality that complements textual information. To extract robust behavioral features, we repurpose established sequence learning models (e.g., DKT (Piech et al., 2015), AKT (Ghosh et al., 2020)) as the Sequence Encoder. Unlike LLMs, these models are specifically optimized to capture sequential dependencies and knowledge state transitions.

Let $\mathcal{E}_{\text{seq}}(\cdot)$ denote the sequence encoder with a hidden dimension d^s. We define the input sequence for a specific interaction attribute as x_k, where $k \in \{q, c\}$ corresponds to questions (QID) and concepts (CID), respectively. The encoder extracts the raw behavioral embeddings $\mathbf{r}_k^{\text{seq}} \in \mathbb{R}^{d^s}$ as follows:

$$\mathbf{r}_k^{\text{seq}} = \mathcal{E}_{\text{seq}}(x_k), \quad \forall k \in \{q, c\} \tag{5}$$

This formulation enables the model to distill complex sequential patterns from discrete ID inputs.

2.3.2 Sequence Adapter

The latent space of traditional KT models differs significantly from the semantic space of LLMs, creating a feature misalignment issue. To resolve this, we employ a Sequence Adapter, parameterized as a projection function $\mathcal{P}_\theta(\cdot)$, to map the behavioral embeddings into a compatible manifold for the LLM. This alignment mechanism is crucial for ensuring that the sequential insights are effectively integrated without disrupting the LLM's reasoning process. The projection is formulated as:

$$\mathbf{h}_k^{\text{seq}} = \mathcal{P}_\theta(\mathbf{r}_k^{\text{seq}}), \quad \forall k \in \{q, c\} \tag{6}$$

where $\mathbf{h}_k^{\text{seq}} \in \mathbb{R}^{d^e}$ means the aligned representations (with d^e being the embedding dimension).

2.4 Baselines

We compare our method with a wide range of baselines across four categories: (1) DL-based approaches—DKT (Piech et al., 2015), DKVMN (Zhang et al., 2017), SAKT (Pandey and Karypis, 2019), AKT (Ghosh et al., 2020), LPKT (Shen et al., 2021), LBKT† (Xu et al., 2023), MRT-KT (Cui et al., 2023) and UKT (Cheng et al., 2025); (2) PLMs-based methods—LBKT‡ (Li et al., 2024), MLFBK (Li et al., 2023), and BiDKT (Tan et al., 2022); (3) Context-aware models—EERNN, EKT (Liu et al., 2019a), RKT (Pandey and Srivastava, 2020), and DCL4KT-A (Lee et al., 2024b); and (4) LLMs-based variants: **LLM-FT**$_{\text{TokenID}}$, **LLM-FT**$_{\text{Text}}$, CLST (Jung et al., 2025), DDKT (Cen et al., 2025) and GPT-4o. For the first three groups, we report results from original or reliable re-implementation papers. For LLM-based methods, **LLM-FT**$_{\text{Text}}$ and GPT-4o are excluded on Assist2015 and Junyi due to the absence of textual question content, and prompts are adapted to available data fields. More details are given in Section C in the Appendix.

3 Experimental Settings

3.1 Datasets and Evaluation

We evaluate our **LLM-KT** model on four standard knowledge tracing benchmarks: ASSISTments2009 (Assist2009) (Ghosh et al., 2020), ASSISTments2015 (Assist2015) (Ghosh et al., 2020), Junyi Academy (Junyi) (bigdata ustc, 2021), and the NeurIPS 2020 Education Challenge (Nips2020)

		Assist2009		Assist2015		Junyi		Nips2020		Avg	
		AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
DL-based Methods	DKT	0.7084	0.7221	0.7093	0.7542	0.8013	0.7200	0.7406	0.6878	0.7399	0.7210
	DKVMN	0.8157	-	0.7268	-	0.8027	-	0.7673	0.7016	0.7781	0.7016
	SAKT	0.8480	-	0.8540	-	0.8340	0.7570	0.7517	0.6879	0.8219	0.7225
	AKT	0.7767	0.7532	0.7211	0.7518	<u>0.8948</u>	0.8215	0.7494	0.6930	0.7855	0.7549
	LPKT	0.7788	0.7325	-	-	0.7689	<u>0.8344</u>	-	-	0.7739	0.7834
	LBKT†	0.7863	0.7380	-	-	0.7723	**0.8362**	-	-	0.7793	0.7871
	AT-DKT	0.7574	0.7172	-	-	0.7581	0.8325	0.7816	0.7145	0.7657	0.7547
	MRT-KT	0.8223	0.7841	-	-	-	-	-	-	0.8223	0.7841
	UKT	0.8563	0.7814	0.7267	0.7497	-	-	0.8035	0.7316	0.7955	0.7542
PLMs-based Methods	BiDKT	0.7651	-	0.6766	-	-	-	-	-	0.7208	-
	MLFBK	<u>0.8524</u>	-	-	-	-	-	-	-	-	-
	LBKT‡	-	-	-	-	0.8510	0.8320	-	-	0.8510	-
Context-Aware Methods	DCL4KT-A	0.8153	-	-	-	-	-	-	-	-	-
	EERNN	-	-	-	-	0.8370	0.7580	-	-	0.8370	0.7580
	EKT	-	-	-	-	0.8420	0.7590	-	-	0.8420	0.7590
	RKT	-	-	-	-	0.8600	0.7700	-	-	0.8600	0.7700
LLMs-based Methods	LLM-FT$_{\text{ID}}$	0.8393	0.7592	<u>0.9092</u>	<u>0.9092</u>	0.8841	0.8071	0.7890	0.6870	<u>0.8554</u>	<u>0.7906</u>
	LLM-FT$_{\text{TokenID}}$	0.8143	0.7954	0.8386	0.8813	0.8663	0.8050	0.7774	0.5962	0.8242	0.7695
	LLM-FT$_{\text{Text}}$	0.8407	<u>0.8119</u>	-	-	-	-	0.7762	<u>0.7211</u>	0.8085	0.7665
	GPT-4o	-	0.7274	-	-	-	-	-	0.6694	-	0.6984
	CLST	-	-	-	-	0.8842	0.8288	0.7535	0.6945	0.8189	0.7617
	DDKT	-	-	-	-	-	-	<u>0.8103</u>	-	0.8103	-
Ours	**LLM-KT**	**0.8870**	**0.8168**	**0.9356**	**0.9185**	**0.9018**	0.8294	**0.8291**	**0.7561**	**0.8884**	**0.8302**

Table 1: Main results of our models and selected baselines. We give the results not reported by the original paper from Cui et al. (2023); Pandey and Srivastava (2020); Piech et al. (2015). Imp. means the relative improvement over the baseline LLM-FT$_{\text{ID}}$. The best and suboptimal results are emphasized in **bold** and <u>underline</u>.

(Wang et al., 2020), with dataset statistics provided in Table 6. Assist2009 and Assist2015 contain student exercise logs with and without question metadata, respectively; Junyi includes over 16 million logs from 72,000+ students over one year; and Nips2020 uses records from the top 150 most frequent questions, with figures manually converted to text. Following prior work, we report performance using AUC and Accuracy (ACC). More details are given in Section B.0.1 in the Appendix.

3.2 Implementation Details

In our experiments, we use the deep learning framework PyTorch Lightning for its ease of use and efficient management of training processes. Following MRT-KT (Cui et al., 2023), we divide the student dataset in a ratio of 8:1:1 for training, validation, and test sets. Then, we train on the training set, select the best model based on the validation set, and evaluate it on the test set. Our **LLM-KT** model is based on LLaMA2 (Touvron et al., 2023). We update our model using the parameter-efficient method LoRA, where the rank is 32, alpha is 32, and dropout is 0.1. Each task is trained for a maximum of 10 epochs with a batch size of 32, using the gradient accumulation strategy. More details are given in Section C.5 in the Appendix.

4 Experimental Analysis

4.1 Main Results

In this section, we report the results of our **LLM-KT** and the selected baselines across four benchmark datasets in terms of AUC and ACC (Table 1). To evaluate the effectiveness of our model, we compare it with four categories: DL-based methods, PLMs-based methods, context-aware methods, and LLMs-based methods.

From these results, we obtain the following observations. **First**, our proposed **LLM-KT** achieves superior performance across all datasets in terms of AUC, demonstrating a clear advantage over existing baselines. In terms of average performance, **LLM-KT** reaches an AUC of 0.8884, surpassing the second-best baseline (**LLM-FT$_{\text{ID}}$**) by a margin of approximately 3.3% (0.8554). Specifically, on the Assist2009 dataset, our model outperforms the competitive DL-based method UKT by 3.07% in AUC (0.8870 vs. 0.8563). **Second**, compared to recent advanced LLM-based approaches such as CLST and DDKT, **LLM-KT** exhibits stronger predictive capabilities. While CLST achieves a respectable AUC of 0.8842 on the Junyi dataset, our model further improves this metric to 0.9018. Similarly, on the Nips2020 dataset, although DDKT shows competitive results (0.8103 AUC), **LLM-KT** still leads by

	Assist2009		Nips2020	
	AUC	ACC	AUC	ACC
LLM-KT	0.8870	0.8168	0.8291	0.7561
Data Source				
- Question	0.8788	0.7937	0.7983	0.7439
- Concept	0.8635	0.8053	0.8101	0.7276
Model Structure				
- Behavior	0.8616	0.8119	0.7958	0.7249
- Semantic	0.8788	0.7937	0.8056	0.7358

Table 2: The results of ablation studies.

1.88% (0.8291 AUC). These comparisons suggest that our architecture is more effective at capturing complex knowledge states than these specialized LLM adaptations. **Third**, simple fine-tuning or direct prompting of LLMs is insufficient for modeling long interaction contexts compared to our approach. For example, directly inputting history records into GPT-4o results in suboptimal accuracy (0.7274 on Assist2009), which is lower than many DL-based baselines. Furthermore, regarding question IDs as tokens in **LLM-FT**$_{\text{TokenID}}$ yields an average AUC of 0.8242, which is noticeably lower than our 0.8884.

4.2 Ablation Studies

To investigate the performance of the main components contained in our proposed **LLM-KT** (Table 2). From the data source, we remove the ID and text of the question (- Question), and the ID and text of the concepts (- Concept) from our model. From the model structure, we remove the Behavioral Dynamics Projector (- Behavior) and Semantic History Projector (- Semantic). Due to missing questions or concepts, we mainly conduct the experiments on the Assist2009 and Nips2020.

From the results, we observe that both question and concept information can help the model understand the student's state from the history record to improve the performance of knowledge tracing. For instance, removing questions from inputs will reduce 3.08 points in terms of AUC (0.8291 vs 0.7983). Additionally, our Behavioral Dynamics Projector and Semantic History Projector effectively capture the sequence interaction behaviors and long textual context. Removing any one of them from our **LLM-KT** will reduce the performance. The textual context helps the model learn the complex semantic relationships between the questions and concepts, and the sequence information helps the model capture the interaction behaviors based on a sequence of IDs.

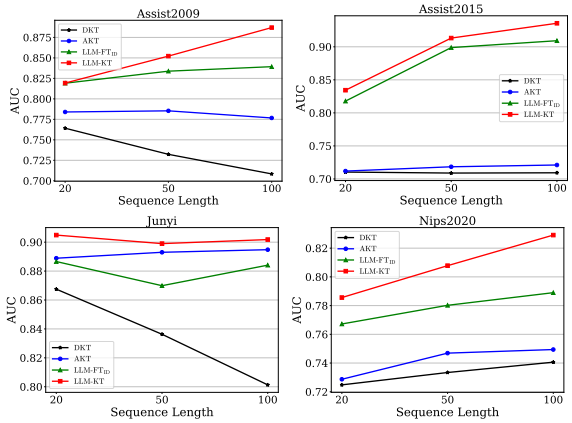

Figure 3: Influence of sequence length on four different datasets in terms of AUC.

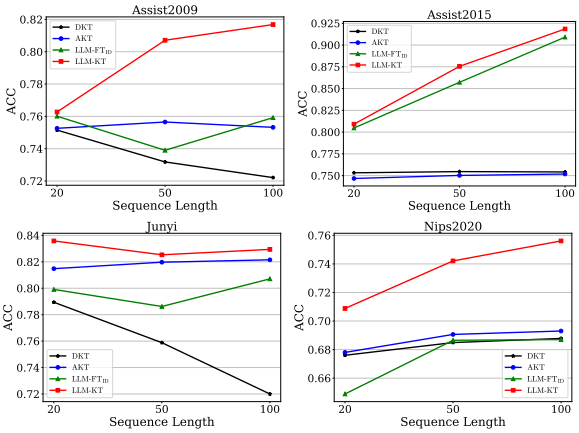

Figure 4: Influence of sequence length on four different datasets in terms of ACC.

4.3 Influence of Sequence Length

In this section, we examine how sequence length affects knowledge tracing. We present our model's results alongside several robust baselines, measured by AUC and ACC (Figure 3 and 4). We define the sequence lengths as 20, 50, or 100.

From the results, we find that our model outperforms the other methods across all sequence lengths, which indicates that **LLM-KT** can capture the students' states with long text and ID sequences effectively. Specifically, our model improves by more than 4 points of AUC compared to DKT and AKT on Nips2020. With the increase in sequence length, our model can effectively improve performance in most cases, while the improvements in previous studies are limited and may even decline. For example, **LLM-KT** with a sequence length of 100 outperforms the one with 20 questions by about 10 points in terms of AUC over Assist2015. The Junyi dataset primarily focuses on short-term records, and extending the sequence length may reduce performance due to unrelated questions in the history

	Assist2009		Assist2015		Junyi		Nips2020		Average	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
LLM-FT$_{\text{TokenID}}$	0.8143	0.7954	0.8386	0.8813	0.8663	0.8050	0.7774	0.5962	0.8242	0.7695
DKT	0.8759	**0.8218**	0.9350	**0.9204**	**0.9027**	**0.8303**	0.8149	0.7453	0.8821	0.8295
AKT	**0.8870**	0.8168	**0.9356**	0.9185	0.9018	0.8294	**0.8291**	**0.7561**	**0.8884**	**0.8302**

Table 3: Influence of sequence encoder. Average means the average score over four datasets.

	Assist2009		Nips2020		Average	
	AUC	ACC	AUC	ACC	AUC	ACC
BERT	0.8770	0.7987	0.8116	0.7331	0.8443	0.7659
MPNET	0.8749	0.8135	0.8231	**0.7561**	0.8490	0.7848
LLaMA2	**0.8870**	**0.8168**	**0.8291**	**0.7561**	**0.8581**	**0.7865**

Table 4: Influence of context encoder.

	Assist2009		Nips2020		Average	
	AUC	ACC	AUC	ACC	AUC	ACC
Concat	0.8851	0.8102	0.8168	0.7466	0.8510	0.7784
Avg	0.8763	**0.8185**	0.8262	0.7466	0.8513	0.7826
Add	**0.8870**	0.8168	**0.8291**	**0.7561**	**0.8581**	**0.7865**

Table 5: Influence of function g.

record. Though our model experiences a performance drop when increasing the length, the decrease is subtle. For DKT, it is an LSTM-based model that is not effective at capturing long sequences. We would like to explore how to reduce the influence of unrelated information in lengthy sequences in the future.

4.4 Further Analysis

4.4.1 Influence of Context Encoder

We investigate the influence of different context encoders and provide their average performance for both AUC and ACC (Table 4). We select three typical sentence encoders: LLaMA2, BERT, and all-mpnet-base-v2 (MPNET). We input the entire question and concepts into the Context-Encoder to generate vectors r_{QText} and r_{CText}. Specifically, the vectors are derived by averaging the hidden states from the last layer along the sequence length for BERT and LLaMA2, while for MPNET we calculate the vector using SentenceTransformer. The dimensions d^t are 768, 768, and 4096 for BERT, MPNET, and LLaMA2, respectively. We develop a Context-Adapter to convert d^t into the embedding layer dimension d^e of LLaMA2 (4096).

Among the models, LLaMA2 achieves the highest average AUC (0.8581) and ACC (0.7865), outperforming BERT by 2 percentage points in ACC (0.7865 vs 0.7659). By training on a large-scale corpus, LLaMA2 learns powerful text embeddings for capturing long textual knowledge. Based on the experiments, we recommend using MPNET, which achieves competitive performance to LLaMA2 while maintaining the same size as BERT.

4.4.2 Influence of Sequence Encoder

We investigate the influence of different sequence encoders and compare them with LLM-FT$_{\text{TokenID}}$ over four datasets (Table 3). We employ DKT (Piech et al., 2015) and AKT (Ghosh et al., 2020) to model the sequence of IDs. For LLM-FT$_{\text{TokenID}}$, it randomly initializes the embeddings of question- and concept-specific tokens and updates them through fine-tuning. In contrast, we use representations learned by DKT and AKT to initialize the token embeddings. The findings indicate that DKT and AKT significantly outperform LLM-FT$_{\text{TokenID}}$, with AKT achieving over 15-point improvement in ACC on Nips2020. This suggests that embeddings from traditional sequence models capture extensive semantic and interaction knowledge, while LLMs cannot capture such knowledge well by simply fine-tuning on ID sequences. Additionally, DKT and AKT obtain comparable results (0.8295 vs 0.8302 for average ACC).

4.4.3 Influence of Function g

To combine the representations of context and sequence, we use a function g to merge them, as mentioned in Equation 1. In our experiment, we explore the influence of different methods (Table 5), including concatenation (Concat), average (Avg) and addition (Add). We find that the impact of the different functions g is limited and the results of these operations are similar. In addition, we validate the generalizability of LLM-KT across different LLM backbones (Appendix D) and provide a detailed cost–performance analysis (Appendix E).

5 Conclusions and Further Work

In this paper, we introduced **LLM-KT**, a novel framework that redefines knowledge tracing by synergizing the reasoning prowess of Large Language Models with the sequential modeling efficacy of traditional methods. Through a Multi-level Plug-and-Play Alignment mechanism, we

effectively translated the specific KT task into a generative language modeling problem, enabling the seamless integration of heterogeneous modalities—specifically, deep semantic contexts and sequential interaction dynamics. Extensive empirical results across four benchmark datasets demonstrate that **LLM-KT** establishes a new state-of-the-art, significantly outperforming competitive baselines. Furthermore, ablation studies validate the critical role of our dual-projector design in capturing both the nuanced semantics of questions and the evolving behavioral patterns of students. Looking ahead, we plan to investigate advanced mechanisms for handling ultra-long interaction sequences, focusing on filtering noise and extracting salient features from extensive learning histories to enhance long-term prediction accuracy.

6 Limitations

While our LLM-KT framework demonstrates superior performance, we acknowledge certain trade-offs inherent to incorporating Large Language Models. Specifically, the hybrid architecture incurs higher computational costs and inference latency compared to lightweight traditional methods, and its full potential relies on the availability of high-quality textual data. Additionally, handling noise in extremely long sequences and generalizing to completely unseen questions remain areas for further optimization. These limitations, however, do not undermine the validity of the proposed mechanism; rather, they stem from current hardware and data constraints, pointing to valuable future research directions in designing more efficient, lightweight architectures and robust long-range attention mechanisms.

Acknowledgements

This research is funded by the National Key Research and Development Program of China Grant (No. 2024YFC3308500), the National Nature Science Foundation of China (No.62477010, No.62577022 and No.62307028), the Natural Science Foundation of Shanghai (No.23ZR1441800), Shanghai Science and Technology Innovation Action Plan (No.24YF2710100 and No.23YF1426100), and CIPS-SMP-Zhipu Large Model Fund.

References

Ghodai Abdelrahman, Qing Wang, and Bernardo Nunes. 2023. Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):1–37.

bigdata ustc. 2021. Edudata. https://github.com/bigdata-ustc/EduData.

Jiahui Cen, Jianghao Lin, Weixuan Zhong, Dong Zhou, Jin Chen, Aimin Yang, and Yongmei Zhou. 2025. Llm-driven effective knowledge tracing by integrating dual-channel difficulty. *Preprint*, arXiv:2502.19915.

Song Cheng, Qi Liu, Enhong Chen, Kai Zhang, Zhenya Huang, Yu Yin, Xiaoqing Huang, and Yu Su. 2022. Adaptkt: A domain adaptable method for knowledge tracing. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 123–131.

Weihua Cheng, Hanwen Du, Chunxiao Li, Ersheng Ni, Liangdi Tan, Tianqi Xu, and Yongxin Ni. 2025. Uncertainty-aware knowledge tracing. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Albert T. Corbett and John R. Anderson. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction*, page 253–278.

Chaoran Cui, Yumo Yao, Chunyun Zhang, Hebo Ma, Yuling Ma, Zhaochun Ren, Chen Zhang, and James Ko. 2024. Dgekt: A dual graph ensemble learning method for knowledge tracing. *ACM Trans. Inf. Syst.*, 42(3).

Jiajun Cui, Zeyuan Chen, Aimin Zhou, Jianyong Wang, and Wei Zhang. 2023. Fine-grained interaction modeling with multi-relational transformer for knowledge tracing. *ACM Transactions on Information Systems*, 41(4):1–26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2330–2339.

Bert F. Green. A general solution for the latent class model of latent structure analysis. *Psychometrika*, 16(2):151–166.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

HuEdward J., Yulong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv: Computation and Language,arXiv: Computation and Language*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Heeseok Jung, Jaesang Yoo, Yohaan Yoon, Yeonju Jang, and 1 others. 2025. Clst: Cold-start mitigation in knowledge tracing by aligning a generative language model as a students' knowledge tracer. *Journal of Educational Data Mining*, 17(2):86–117.

Unggi Lee, Jiyeong Bae, Dohee Kim, Sookbun Lee, Jaekwon Park, Taekyung Ahn, Gunho Lee, Damji Stratton, and Hyeoncheol Kim. 2024a. Language model can do knowledge tracing: Simple but effective method to integrate language model and knowledge tracing task. *arXiv preprint arXiv:2406.02893*.

Unggi Lee, Sungjun Yoon, Joon Seo Yun, Kyoungsoo Park, Younghoon Jung, Damji Stratton, and Hyeoncheol Kim. 2024b. Difficulty-focused contrastive learning for knowledge tracing with a large language model-based difficulty prediction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 4891–4900. ELRA and ICCL.

Haoxuan Li, Jifan Yu, Yuanxin Ouyang, Zhuang Liu, Wenge Rong, Huiqin Liu, Juanzi Li, and Zhang Xiong. 2025. Explainable few-shot knowledge tracing. *Frontiers of Digital Education*, 2(4):34.

Zhaoxing Li, Mark Jacobsen, Lei Shi, Yunzhan Zhou, and Jindi Wang. 2023. Broader and deeper: A multi-features with latent relations bert knowledge tracing model. In *Responsive and Sustainable Educational Futures*, pages 183–197, Cham. Springer Nature Switzerland.

Zhaoxing Li, Jujie Yang, Jindi Wang, Lei Shi, and Sebastian Stein. 2024. Integrating lstm and bert for long-sequence data analysis in intelligent tutoring systems. *arXiv preprint arXiv:2405.05136*.

Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kaylan. 2023. Let gpt be a math tutor: Teaching math word problem solvers with customized exercise generation. *Preprint*, arXiv:2305.14386.

Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019a. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yunfei Liu, Yang Yang, Xianyu Chen, Jian Shen, Haifeng Zhang, and Yong Yu. 2021. Improving knowledge tracing via pre-training question embeddings. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1577–1583.

Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The world wide web conference*, pages 3101–3107.

Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*.

Seyed Parsa Neshaei, Richard Lee Davis, Adam Hazimeh, Bojan Lazarevski, Pierre Dillenbourg, and Tanja Käser. 2024. Towards modeling learner performance with large language models. *arXiv preprint arXiv:2403.14661*.

Lin Ni, Sijie Wang, Zeyu Zhang, Xiaoxuan Li, Xianda Zheng, Paul Denny, and Jiamou Liu. 2024. Enhancing student performance prediction on learnersourced questions with sgnn-llm synergy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23232–23240.

Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.

Shalini Pandey and Jaideep Srivastava. 2020. Rkt: Relation-aware self-attention for knowledge tracing. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.

Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. 2021. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1452–1460.

Shuanghong Shen, Qi Liu, Zhenya Huang, Yonghe Zheng, Minghao Yin, Minjuan Wang, and Enhong Chen. 2024. A survey of knowledge tracing: Models, variants, and applications. *IEEE Transactions on Learning Technologies*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Xiangyu Song, Jianxin Li, Qi Lei, Wei Zhao, Yunliang Chen, and Ajmal Mian. 2022. Bi-clkt: Bi-graph contrastive learning based knowledge tracing. *Knowledge-Based Systems*, 241:108274.

Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jianwen Sun, Mengqi Wei, Jintian Feng, Fenghua Yu, Qing Li, and Rui Zou. 2024. Progressive knowledge tracing: Modeling learning process from abstract to concrete. *Expert Systems with Applications*, 238:122280.

Weicong Tan, Yuan Jin, Ming Liu, and He Zhang. 2022. Bidkt: Deep knowledge tracing with bert. In *Ad Hoc Networks and Tools for IT*, pages 260–278, Cham. Springer International Publishing.

Zejie Tiana, Guangcong Zhengc, Brendan Flanaganb, Jiazhi Mic, and Hiroaki Ogatab. 2021. Bekt: deep knowledge tracing with bidirectional encoder representations from transformers. In *Proceedings of the 29th International Conference on Computers in Education*, volume 2, pages 6–2.

Hanshuang Tong, Yun Zhou, and Zhen Wang. 2020. Exercise hierarchical feature enhanced knowledge tracing. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 324–328. Springer.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Ashish Vaswani. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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

Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, Simon Woodhead, and Cheng Zhang. 2020. Diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*.

Jianghua Xia, Han Wang, Qingfeng Zhuge, and Edwin Hsing-Mean Sha. 2023. Knowledge tracing model and student profile based on clustering-neural-network. *Applied Sciences*, 13(9):5220.

Bihan Xu, Zhenya Huang, Jiayu Liu, Shuanghong Shen, Qi Liu, Enhong Chen, Jinze Wu, and Shijin Wang. 2023. Learning behavior-oriented knowledge tracing. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2789–2800.

Andrea Zanellati, Daniele Di Mitri, Maurizio Gabbrielli, and Olivia Levrini. 2024. Hybrid models for knowledge tracing: A systematic literature review. *IEEE Transactions on Learning Technologies*.

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

Appendix

A Related Work

A.1 Deep learning-based Knowledge Tracing

Traditional knowledge tracking algorithms are mainly based on machine learning algorithms, such as Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1995) and Item Response Theory (IRT) (Green). With the continuous development and progress of neural networks, deep learning-based knowledge tracing algorithms have emerged to model the sequence interaction (Piech et al., 2015; Cui et al., 2023; Shen et al., 2021; Cheng et al., 2022). DKT (Piech et al., 2015), or Deep Knowledge Tracing, is the first model to apply deep learning to the field of knowledge tracing, which learns the features of students' historical problem-solving records using Long Short-Term Memory (LSTM). SAKT (Pandey and Karypis, 2019) utilized the self-attention mechanism to address the problem of insufficient generalization ability existing in the processing of sparse data. AKT (Ghosh et al., 2020) further introduced a new monotonic attention mechanism and the classic Rasch-model in psychometrics to better understand students' knowledge mastery status and learning processes. BEKT (Tiana et al., 2021) proposed a multi-layer bidirectional transformer encoder with a self-attention mechanism and bidirectional analysis, to understand the student's past learning logs. Zhang

et al. (2017) proposed a new structure called Dynamic Key-Value Memory Networks (DKVMN), which can utilize the relationships between underlying concepts and directly output the mastery level of each concept by students.

To further evaluate the time aspect, DKT-Forget (Nagatani et al., 2019) enhances DKT by translating the time interval into a numerical value. This value, along with learning interaction data like answering questions, is fed into the neural network. In contrast, HawkesKT (Wang et al., 2021) leverages the intensity function and mechanisms of the Hawkes process to measure the triggering effects of events across different time points. This approach clarifies how learning events temporally influence the probability of subsequent occurrences and the knowledge state. Addressing limitations in the learning process, which is vital for KT tasks, LPKT (Shen et al., 2021) assesses students' knowledge states by modeling their learning journey, capturing knowledge gains while also considering the phenomenon of forgetting. Simultaneously, Sun et al. (2024) offers a novel perspective in the KT field by developing the Progressive Knowledge Tracing model. This model emphasizes the learning journey through students' sequential thought processes and divides it into three relatively independent, yet progressively advanced stages: concept mastery, question-solving, and answering behavior, effectively modeling the transition from abstract reasoning to concrete responses.

Furthermore, graph neural networks are used to model the relationships between different questions or knowledge points in the field of knowledge tracing (Nakagawa et al., 2019; Song et al., 2022; Liu et al., 2021; Cui et al., 2024). GKT (Nakagawa et al., 2019) constructs a knowledge graph based on knowledge points or questions, and utilizes Graph Neural Networks (GNNs) to explore and take advantage of these underlying relational structures. BI-CLKT (Song et al., 2022) designs a two-layer comparative learning scheme on an "exercise-to-exercise" (E2E) relational subgraph for node-level and graph-level contrastive learning to get discriminative representations of exercises and concepts. Additionally, two variants with different prediction layers (RNN and memory-augmented neural networks) are explored to improve representations. PEBG (Liu et al., 2021) puts forward a pre-training embedding method through a bipartite graph (PEBG), leveraging edge information (including question difficulty, explicit question-skill relationships, implicit question similarity, and skill similarity) to learn low-dimensional embeddings for each question. DGEKT (Cui et al., 2024) innovatively constructs a dual graph structure of students' learning interactions, using a concept association hypergraph and a directed transition graph to capture heterogeneous relationships. Additionally, it employs online knowledge distillation to adaptively combine the dual graph models, forming a stronger ensemble teacher model for enhanced modeling ability.

A.2 PLMs-Enhanced Knowledge Tracing

In the field of knowledge tracing, Pre-trained Language Models (PLMs), such as BERT (Devlin et al., 2019), RoBERT (Liu et al., 2019b), are used to enhance the semantic representation for knowledge tracing (Tan et al., 2022; Li et al., 2023; Tong et al., 2020; Lee et al., 2024b). For instance, BiDKT (Tan et al., 2022) adapts BERT to trace knowledge by predicting the correctness of randomly masked responses within sequences. MLFBK (Li et al., 2023) leverages the power of BERT to mine latent relations among multiple explicit features, such as individual skill mastery, students' ability profiles, and problem difficulty. Furthermore, Tong et al. (2020) proposes a hierarchical exercise feature enhanced knowledge tracing framework that utilizes BERT to generate exercise text embeddings and then feeds them into three systems to extract knowledge distribution, semantic features, and difficulty features. Moreover, LBKT (Li et al., 2024) combines the strengths of BERT for capturing complex data relations and LSTM for handling long sequences, enhancing performance on data with over 400 interactions. However, the integration of LLMs with knowledge tracing has not been explored well.

A.3 Context-aware Knowledge Tracing

The context information (such as textual features in the questions and concepts) contains a wealth of semantic knowledge, which can help reduce the cold-start phenomenon of KT. Several studies utilized the context information to enhance traditional deep learning models (Pandey and Srivastava, 2020; Su et al., 2018). For example, RKT (Pandey and Srivastava, 2020) used the textual information of the questions to capture relations between exercises. EERNN (Su et al., 2018) and EKT (Liu et al., 2019a) considered the text of the questions to learn a good question representation for knowledge tracing. Additionally, Liu et al. (2021) proposed

a pre-training method called PEBG, which learns question embeddings with rich relational information using the bipartite graph of question-skill relations. Moreover, Lee et al. (2024b) proposed a difficulty-centered contrastive learning method based on the question representations using BERT.

Unlike previous research, we propose **LLM-KT** to combine the great advantages of LLMs and traditional sequence learning models for KT. We design a plug-and-play instruction to align the context and sequence representations with LLMs.

B Datasets

B.0.1 Datasets

To evaluate the effectiveness of our **LLM-KT**, we conduct experiments on four commonly used benchmark datasets for knowledge tracing. The statistical information of these datasets is listed in Table 6.

- ASSISTments2009 (Assist2009) (Ghosh et al., 2020) collects the exercises of 4151 students during the 2009 to 2010 school year. The same as Ghosh et al. (2020), we use the skill builder data version of this dataset. To ensure the validity of the data, we only retain those records where both the skill_name and skill_id fields are not empty.

- ASSISTments2015 (Assist2015) (Ghosh et al., 2020) comprises responses from students on 100 distinct questions. Different from Assist2009, this dataset does not provide metadata of questions.

- Junyi Academy (Junyi) (bigdata ustc, 2021) is provided by Junyi Academy - the premier online learning platform in Taiwan, consisting of over 16 million exercise attempt logs. These logs are contributed by more than 72,000 students, spanning a year, specifically from August 2018 to July 2019. We use the dataset provided by bigdata ustc (2021), which is processed specifically for knowledge tracing.

- NeurIPS 2020 Education Challenge (Nips2020) (Wang et al., 2020) is released by the NeurIPS 2020 Education Challenge. In this paper, we use the datasets from Challenge Task 3 & 4 and extract the records of the top 150 most frequently appearing questions. Note that, to obtain the textual information of the questions, we convert the figures into text manually.

C Baselines

In our research, we compare our proposed approach with several strong baseline methodologies to assess its efficacy and performance. We split these baselines into four parts: deep-learning (DL)-based, pre-trained language models (PLMs)-based, context-aware and LLMs-based methods.

C.1 DL-based Methods

DL-based methods learn the interactions among students' records effectively by taking the relationships and times into account. Here, we select 7 typical baselines as follows:

- **DKT** (Piech et al., 2015) uses RNNs((Elman, 1990) to model temporal dependencies in student learning, capturing the evolution of knowledge states.

- **DKVMN** (Zhang et al., 2017) implements a dynamic key-value memory network, where static matrices store knowledge concepts and dynamic matrices update mastery levels, enhancing the modeling of concept relationships.

- **SAKT** (Pandey and Karypis, 2019) employs a self-attention mechanism((Vaswani, 2017)) to identify key knowledge concepts (KCs) from past interactions.

- **AKT** (Ghosh et al., 2020) utilizes a monotonic attention mechanism to build context-aware representations of student interactions, capturing performance over appropriate time scales.

- **LPKT** (Shen et al., 2021) models the learning process by formalizing learning cells and incorporating gates for managing retention and forgetting over time.

- **LBKT**† (Xu et al., 2023) analyzes the interplay of learning behaviors (e.g., speed, attempts, hints) and uses a forgetting factor to update learners' knowledge states.

- **MRT-KT** (Cui et al., 2023) employs a multi-relational transformer with a novel relation encoding scheme to model fine-grained interactions between question-answer pairs in knowledge tracing.

- **UKT** (Cheng et al., 2025) employs stochastic distribution embeddings to represent uncertainty in student interactions, utilizing a Wasserstein

Dataset	Students	Questions	KCs	Interactions	QID	CID	QuesCont	ConcCont
Assist2009	4,151	16,891	110	325,637	✓	✓	✗	✓
Assist2015	19,840	-	100	683,801	✗	✓	✗	✗
Junyi	1,000	834	-	972,855	✓	✗	✗	✗
Nips2020	5,310	110	17	428,596	✓	✓	✓	✓

Table 6: The statistical information of the datasets. QID and CID mean the ID of the question and concept. KCs means the number of knowledge concepts. QuesCont and ConcCont represent the context of the question and concept.

self-attention mechanism to capture state transitions and incorporating aleatory uncertainty-aware contrastive learning to enhance robustness.

C.2 PLMs-based Methods

PLMs-based methods improve the performance of knowledge tracing via the rich knowledge and powerful natural language understanding of PLMs. Here, we adopt the following baselines:

- **LBKT**‡ (Li et al., 2024) addresses long-sequence data in knowledge tracing by integrating a BERT-based architecture with Rasch model embeddings for difficulty levels and an LSTM for sequential processing.
- **MLFBK** (Li et al., 2023) utilizes BERT to incorporate explicit features and latent relations, enhancing prediction efficiency in knowledge tracing.
- **BiDKT** (Tan et al., 2022) adapts BERT for knowledge tracing by leveraging bidirectional context in interaction histories, unlike traditional RNN-based models.

C.3 Context-Aware Methods

For context-aware methods, they utilize the context of questions to learn semantic knowledge. Particularly, we select the following four algorithms:

- **EERNN** (Liu et al., 2019a) combines student records and exercise content into a single vector, processed by a bidirectional LSTM, with two variants: EERNNM (Markov property) and EERNNA (Attention mechanism).
- **EKT** (Liu et al., 2019a) extends EERNN by using a knowledge state matrix, which captures the impact of exercises on multiple concepts, while a memory network tracks concept mastery.
- **RKT** (Pandey and Srivastava, 2020) uses relation-aware self-attention to integrate contextual information from exercises and performance data. It also includes a forgetting model with an exponentially decaying kernel to address interactions and forgetfulness.
- **DCL4KT-A** (Lee et al., 2024b) introduces a difficulty-centered contrastive learning method and leverages LLMs to optimize and predict difficulty from unseen data.

C.4 LLMs-based Methods

Furthermore, LLM-based methods have good reasoning abilities with rich commonsense knowledge. We conduct four versions of LLMs based on both fine-tuning and prompting:

- **LLM-FT**$_{\text{ID}}$ finetunes the LLaMA model with instructions using QIDs and/or CIDs depending on the dataset.
- **LLM-FT**$_{\text{TokenID}}$ finetunes the LLaMA model by treating QID and CID as specific tokens, where their embeddings are updated during training.
- **LLM-FT**$_{\text{Text}}$ finetunes the LLaMA model using textual information of questions and concepts.
- **GPT-4o** inputs the same textual information as LLM-FT$_{\text{Text}}$ directly into the GPT-4o framework.
- **CLST** (Jung et al., 2025) mitigates cold-start problems in knowledge tracing by aligning a generative language model as a students' knowledge tracer.
- **DDKT** (Cen et al., 2025) leverages LLMs and Retrieval-Augmented Generation (RAG) to assess question difficulty from multiple perspectives and model student mastery levels through difficulty-aware mechanisms.

For DL-based, PLMs-based, and Context-aware Methods, we only report the results from the original papers or other relevant experimental papers to ensure the reliability of the experimental results.

For LLMs-based Methods, since Assist2015 and Junyi have no textual information of the question and concept, we don't provide the results of LLM-FT$_{Text}$ and GPT-4o. Note that we remove the information missed in the corresponding dataset (such as QID in Assist2015) from the prompt template in our experiments.

C.5 Implementation Details

In our experiments, we use the deep learning framework PyTorch Lightning for its ease of use and efficient management of training processes. Following MRT-KT (Cui et al., 2023), we divide the student dataset in a ratio of 8:1:1 for training, validation, and test sets. Then, we train on the training set, select the best model based on the validation set, and evaluate it on the test set. Our **LLM-KT** model is based on LLaMA2 (Touvron et al., 2023), which is an advanced and commonly used open-source LLM with over 9,000 citations. We update our model using the parameter-efficient method LoRA, where the rank is 32, alpha is 32, and dropout is 0.1. Each task is trained for a maximum of 10 epochs with a batch size of 32, using the gradient accumulation strategy. For the function g used to merge the representations of context and sequence, we employ the addition operation. We utilize early stopping to avoid overfitting. Additionally, we use Adam as the optimizer with a learning rate of 3×10^{-4} and a weight decay of 1×10^{-5}, utilizing a cosine learning rate scheduler. The sequence length of historical records is 100. We adopt LLaMA2-7B as the context encoder and AKT as the sequence encoder.

To elucidate the distinctions among diverse models within LLMs-based methods more perspicuously, we initially streamline the template in Section 2 as follows.

Input x.
The student has previously, in chronological order, answered HistoryQues$_1$, HistoryQues$_2$,..., HistoryQues$_n$. Please predict whether the student will answer TargetQues correctly. Response with 'Yes' or 'No'. Response:

For LLM-FT$_{\text{ID}}$, we simply replace each "HistoryQues" in the template with the format "question with ID=QID involving concept ID=CID correctly." At the same time, we change "TargetQues" to "the next question with ID=QID involving concept ID=CID correctly." Note that the actual values of these IDs depend on the specific responses shared by students. For LLM-FT$_{\text{TokenID}}$, we adjust "HistoryQues" in the template with "question with ID=[qid74] involving concept ID=[cid6] correctly" and turn "TargetQues" into "the next question with ID=[qid44] involving concept ID=[cid5] correctly." Here, [qid74]/[cid6]/[qid44]/[cid5] represents a newly introduced token that is finetuned on the target dataset to learn the semantic and interaction information in the given record. The number of newly added QID/CID tokens exactly matches the number of questions and concepts. As for LLM-FT$_{\text{Text}}$ and GPT-4o, we replace "HistoryQues" and "TargetQues" in the template with specific textual questions. For example, we use "Which symbol belongs in the box? Pic$_{749-0}$ A:$>$ B:$<$ C:$=$ D:$\geq$ Related knowledge concepts: Basic Arithmetic The student answered this question correctly." We then input these into LLaMA and GPT-4o, respectively.

D LLM Backbone Generalizability

A key design principle of LLM-KT is that the alignment modules are *structurally independent* of the underlying LLM backbone. To validate this, we replace LLaMA2-7B with Mistral-7B (Jiang et al., 2023) while keeping all other components unchanged (sequence encoder, context encoder, projector architectures, and LoRA configuration). As shown in Table 11, Mistral-7B achieves competitive performance (average AUC 0.8843 vs. 0.8884 for LLaMA2-7B), with only marginal differences on individual datasets. This confirms that LLM-KT's effectiveness stems from the multi-level alignment mechanism rather than a specific backbone, and practitioners can freely substitute the LLM component based on availability or computational budget.

E Cost–Performance Analysis

We provide a cost–performance analysis to transparently characterize the computational overhead of LLM-KT. All measurements are conducted on the Nips2020 dataset using a single NVIDIA A800 GPU, with batch size 1 and sequence length 100.

As shown in Table 12, incorporating an LLM backbone inevitably incurs higher costs than lightweight methods: LLM-KT requires 2706 s per training epoch and 350 ms per inference sample, significantly slower than DKT (674 s, 16.6 ms) and AKT (172 s, 3.6 ms). Compared to LLM-FT$_{\text{ID}}$,

Type	Template
1	The student has previously, in chronological order, answered question with ID=74 [WrapQEmb] involving concept ID=6 [WrapCEmb] correctly, ..., question with ID=42 [WrapQEmb] involving concept ID=5 [WrapCEmb] incorrectly. Please predict whether the student will answer the next question with ID=44 [NextWrapQEmb] involving concept ID=5 [NextWrapCEmb] correctly. Response with 'Yes' or 'No'.
2	The student has previously, in chronological order, answered question with ID=3117 [QidEmb] correctly, question with ID=2964 [QidEmb] correctly, question with ID=5627 [QidEmb] incorrectly, ..., question with ID=5532 [QidEmb] correctly. Please predict whether the student will answer the next question with ID=5707 [NextQidEmb] correctly. Response with 'Yes' or 'No'.
3	The student has previously, in chronological order, answered question involving concept ID=15 [CidEmb] correctly, ..., question involving concept ID=30 [NextCidEmb] correctly. Please predict whether the student will answer the next question involving concept ID=30 correctly. Response with 'Yes' or 'No'.

Table 7: The prompt templates for LLM-KT(Ours)

Type	Template
1	The student has previously, in chronological order, answered question with ID=74 involving concept ID=6 correctly, question with ID=80 involving concept ID=6 correctly, ..., question with ID=42 involving concept ID=5 incorrectly. Please predict whether the student will answer the next question with ID=44 involving concept ID=5 correctly. Response with 'Yes' or 'No'.
2	The student has previously, in chronological order, answered question with ID=3117 correctly, question with ID=2964 correctly, ..., question with ID=5532 correctly. Please predict whether the student will answer the next question with ID=5707 correctly. Response with 'Yes' or 'No'.
3	The student has previously, in chronological order, answered question involving concept ID=15 correctly, question involving concept ID=15 correctly, ..., question involving concept ID=30 correctly. Please predict whether the student will answer the next question involving concept ID=30 correctly. Response with 'Yes' or 'No'.

Table 8: The prompt templates for LLM-FT_{ID}

Type	Template
1	The student has previously, in chronological order, answered question with ID=[qid_{74}] involving concept ID=[cid_{6}] correctly, question with ID=[qid_{80}] involving concept ID=[cid_{6}] correctly, ..., question with ID=[qid_{42}] involving concept ID=[cid_{5}] incorrectly. Please predict whether the student will answer the next question with ID=[qid_{44}] involving concept ID=[cid_{5}] correctly. Response with 'Yes' or 'No'.
2	The student has previously, in chronological order, answered question with ID=[qid_{3117}] correctly, question with ID=[qid_{2964}] correctly, question with ID=[qid_{5627}] incorrectly, ..., question with ID=[qid_{5532}] correctly. Please predict whether the student will answer the next question with ID=[qid_{5707}] correctly. Response with 'Yes' or 'No'.
3	The student has previously, in chronological order, answered question involving concept ID=[cid_{15}] correctly, question involving concept ID=[cid_{15}] correctly, question involving concept ID=[cid_{30}] incorrectly, ..., question involving concept ID=[cid_{30}] correctly. Please predict whether the student will answer the next question involving concept ID=[cid_{30}] correctly. Response with 'Yes' or 'No'.

Table 9: The prompt templates for LLM-$\text{FT}_{\text{TokenID}}$

which shares the same backbone, the projector modules (51.4 M, $<$0.8% of total parameters) add moderate overhead (+26.6% training, +14.0% inference). Nevertheless, this cost yields clear gains: LLM-KT achieves 0.8884 average AUC, outperforming DKT by 14.85%, AKT by 10.29%, and LLM-FT_{ID} by 3.3%. The dominant bottleneck is the LLM's autoregressive decoding, shared across all LLM-based methods, and can be mitigated by orthogonal techniques such as quantization.

F Prompt Templates

In this section, we provide detailed descriptions of the prompt templates used for different datasets in our study. These templates are designed to handle various types of data and adapt to the specific requirements of each dataset.

We introduce five distinct prompt templates:

- **Type 1** (Combined Question and Concept ID Template): Used for datasets with both QIDs and CIDs, applicable to Assist2009 and Nips2020.
- **Type 2** (Question ID-Only Template): Used

Type	Template
4	In this task, we aim to determine whether the student can answer the question correctly based on the student's history record of academic exercises. The student's history record of academic exercises is given as follows: 1) How would this calculation be written? Pic_{290-0} A:8+(2÷5)=2 B:(8+2)÷5=2 C:8+2÷5=2 D:(8+2÷5)=2 Related knowledge concepts: Basic Arithmetic The student answered this question correctly 2) Which symbol belongs in the box? Pic_{749-0} A:$>$ B:$<$ C:$=$ D:$\geq$ Related knowledge concepts: Basic Arithmetic The student answered this question correctly 3) What is the output of this Function Machine? Pic_{836-0} A:10p B:7p C:5(p+2) D:5p+2 Related knowledge concepts: Writing Expressions The student answered this question incorrectly The target question is given as follows: Tom and Katie are arguing about the result of this Function Machine: Pic_{856-0}. Tom says the output is: 3n-12. Katie says the output is:3(n-4). Who is correct? A:Only Tom B:Only Katie C:Both Tom and Katie D:Neither is correct Related knowledge concepts: Writing Expressions Please predict whether the student would answer the target question correctly. Response with 'Yes' or 'No'.
5	The student has previously, in chronological order, answered question involving concept "Basic Arithmetic" correctly, question involving concept "Basic Arithmetic" correctly, ..., question involving concept "Basic Arithmetic" incorrectly, question involving concept "Basic Arithmetic" correctly, ..., question involving concept "Ordering Negative Numbers" incorrectly, question involving concept "Ordering Negative Numbers" correctly. Please predict whether the student will answer the next question involving concept "Ordering Negative Numbers" correctly. Response with 'Yes' or 'No'.

Table 10: The prompt templates for $\text{LLM-FT}_{\text{Text}}$

Table 11: Performance comparison across different LLM backbones. All settings use AKT as the sequence encoder and the same LoRA configuration.

Backbone	Assist2009		Assist2015		Junyi		Nips2020	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
LLaMA2-7B	0.8870	0.8168	0.9356	0.9185	0.9018	0.8294	0.8291	0.7561
Mistral-7B	0.8831	0.8195	0.9312	0.9147	0.8976	0.8312	0.8254	0.7498

exclusively for datasets with only QIDs, such as Junyi.

- **Type 3** (Concept ID-Only Template): Used exclusively for datasets with only CIDs, like Assist2015.
- **Type 4** (Contextual Question Template): Used for datasets with text associated with questions, applicable only to Nips2020.
- **Type 5** (Contextual Concept Template): Used for datasets with concept text, like Assist2009.

G Terminology Explanation

- **QID** (Question ID): The unique identifier for each question, used to track and model the sequence of a student's answers.
- **CID** (Concept ID): The unique identifier for the knowledge concept tied to each question.
- **WrapQEmb** (Wrapped Question Embedding): The embedding formed by combining the QID and the question's text, leveraging both identity and semantic content.
- **WrapCEmb** (Wrapped Concept Embedding): Similar to 'WrapQEmb', but combines the CID with the concept's text.
- **QidEmb** (Question ID Embedding): An embedding of the QID, used without the question's text in templates focused on identity.
- **CidEmb** (Concept ID Embedding): An embedding of the CID, used without the concept's text in simpler templates.
- **NextWrapQEmb** (Next Wrapped Question Embedding): The fused embedding for the next QID, combining its ID and text similar to 'WrapQEmb'.

Table 12: Cost–performance comparison on the Nips2020 dataset with batch size set to 1. "Trainable Params" refers to the number of parameters updated during training. "Projector Params" indicates the additional parameters introduced by our alignment modules. Training time is reported per epoch; inference latency is per sample.

Method	Total Params	Trainable Params	Projector Params	Train (s/epoch)	Infer (ms/sample)
DKT	0.43 M	0.43 M	—	674	16.6
AKT	4.18 M	4.18 M	—	172	3.6
LLM-FT_{ID}	6.8 B	40.0 M	—	2138	307
LLM-KT (Ours)	6.8 B	91.4 M	51.4 M	2706	350

- **NextQidEmb** (Next Question ID Embedding): The next QID's embedding, used without the question's text.
- **NextWrapCEmb** (Next Wrapped Concept Embedding): The fused embedding for the next CID, combining its ID and text.
- **NextCidEmb** (Next Concept ID Embedding): The next CID's embedding, used without the concept's text.