GLLM-KT: A Graph-Incorporated Ultra-Small Large Language Model for Knowledge Tracing

Lianhong Wang[1†][0000−0001−8016−3042], Guiyao Liu[1†], Xiaoyao Li[2∗][0000−0002−4600−1215], Junyi Li[1], Xiaogang Zhang[1], and Xiang Yin[1]

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China

[2] School of Advanced Interdisciplinary Studies, Central South University of Forestry and Technology, Changsha 410004, China

lxy0731@csuft.edu.cn

Abstract. Knowledge Tracing (KT) aims to model students' latent knowledge states from historical learning interactions. However, existing KT methods often struggle with generalization across diverse datasets. Furthermore, while Large Language Models (LLMs) have shown potential for improving KT, their enormous parameter sizes limits practical deployment. To address these issues, we propose GLLM-KT, a graph-incorporated ultra-small LLM-based KT model. It integrates a knowledge prerequisite graph to capture concept dependencies, a structured instruction template for task formulation, and an adaptive confidence head to improve prediction robustness. In this paper, we investigate the potential of ultra-small large language models for KT. Extensive experiments on four benchmark datasets demonstrate the superior performance of GLLM-KT in prediction accuracy and generalization across different datasets and scales of ultra-small LLMs.

Keywords: Knowledge Tracing, Large Language Models, Educational Data Mining.

1 Introduction

Knowledge Tracing (KT) aims to model student's evolving knowledge states using their learning interactions. It plays an important role in intelligent tutoring and adaptive learning systems. By estimating student's mastery of knowledge concepts, KT supports personalized guidance, adaptive assessment, and data-driven curriculum design, thereby improving both learning efficiency and educational outcomes [1, 2].

KT research has evolved from early probabilistic models to neural networks, and more recently to paradigms based on Large Language Model (LLM). Classical probabilistic models [1] treat knowledge states as binary latent variables with fixed transition probabilities, providing high interpretability but limited

† These authors contributed equally to this work.

∗ Corresponding author.

flexibility in modeling temporal dynamics. Neural networks-based methods [2, 3] have been developed to capture nonlinear temporal dependencies using recurrent and memory-augmented architectures. In particular, self-attention-based networks [4, 5] significantly improve the modeling of long-time dependencies, while graph-based networks [6, 7] integrate explicit concept relations for structured reasoning. Despite these advancements, most existing methods often struggle to generalize across datasets and diverse educational contexts. Recent progresses in LLMs offer promising avenues for overcoming these limitations in KT. Pretrained on extensive corpora, LLMs exhibit strong contextual comprehension and reasoning abilities. Studies [8, 9] have indicated that LLMs can model student's learning trajectories more effectively than traditional neural networks-based KT models. However, their parameter scale typically ranges from tens to hundreds of billions, leading to high computational costs and thereby hindering practical employment in educational environments.

To address these issues, we adopt the ultra-small LLM Qwen3 [10] as the baseline model and propose a graph-incorporated ultra-small LLM-based KT model (GLLM-KT). Our main contributions are summarized as follows:

- We design a knowledge prerequisite graph construction algorithm to enable GLLM-KT to effectively capture dependencies among knowledge concepts.
- To enhance the interpretability of GLLM-KT, we introduce an Adaptive Confidence Head (ACH) that integrates the hidden feature representations from the ultra-small LLM to produce stable predictions.
- Comprehensive experimental results on four benchmark datasets demonstrate the superiority of GLLM-KT in both effectiveness and generalization across diverse datasets and ultra-small LLMs of varying parameter scales.

2 Related Work

2.1 Probabilistic Knowledge Tracing

Early KT research was primarily based on probabilistic frameworks like Bayesian Knowledge Tracing (BKT) [1], which modeled learning and forgetting behaviors as latent state transitions. Despite its interpretability, the parameter invariance assumption constrains BKT's ability to capture diverse learning dynamics. Although subsequent extensions including time-dependent BKT [16] and performance factor analysis [17] enhance temporal flexibility, they remain heavily dependent on hand-crafted rules.

2.2 Neural Knowledge Tracing

Since the emergence of deep learning, it has drawn increasing attention in KT. Deep Knowledge Tracing (DKT) [2] utilizes Recurrent Neural Networks (RNNs) to learn nonlinear temporal dependencies, while dynamic key-value memory networks [3] incorporates memory modules to explicitly represent knowledge state. Despite strong prediction performance, these models often suffer from overfitting, poor calibration and weak generalization across datasets.

To overcome the limitations of RNNs, self-attention mechanisms have been introduced in KT models like Attentive Knowledge Tracing (AKT) [5] and self attentive knowledge tracing [4]. These approaches can selectively focus on relevant historical learning interactions and incorporate contextual embeddings, thus improving sequence modeling and robustness on sparse data. However, they still struggle to encode dependencies among knowledge concepts. Graph-based methods, such as Graph-based Knowledge Tracing (GKT) [6] and Multiple Graph Knowledge Tracing (MGKT) [7], explicitly model the prerequisite and co-occurrence relationships among knowledge concepts using graph neural networks, enhancing interpretability and structural reasoning. However, the high computational cost incurred by large concept graphs limits their deployment in large-scale or real-time educational systems.

2.3 LLM-Based Knowledge Tracing

With superior contextual reasoning and transfer capabilities, LLMs have emerged as a promising foundation for KT. While studies [8, 9] have demonstrate the feasibility of LLMs to track student's knowledge states through textual prompts and natural language reasoning, their huge parameter size poses significant challenges in terms of high memory requirements. To address above challenge, parameter-efficient fine-tuning methods like Low-Rank Adaptation (LoRA) [11] enable low-cost adaptation with minimal additional parameters.

3 Methodology

In this section, we first provide the problem formulation for KT, and then introduce the proposed GLLM-KT with three components, i.e. knowledge prerequisite graph, structured instruction template and adaptive confidence head for prediction. Figure 1 illustrates the overall flowchart of GLLM-KT.

3.1 Problem Formulation

KT models a student's evolving knowledge state based on their historical learning interactions. Formally, an interaction sequence is defined as:

$$\mathcal{S} = \{(q_1, \mathcal{C}_1, a_1),\ (q_2, \mathcal{C}_2, a_2),\ \ldots,\ (q_{L_m}, \mathcal{C}_{L_m}, a_{L_m})\}, \tag{1}$$

where $q_i \in \mathcal{Q}$, $\mathcal{C}_i \subseteq \mathcal{C}$ and $a_i \in \{0, 1\}$ denote the item index, the associated knowledge concept(s) and the response correctness related to q_i, respectively. $\mathcal{Q}$ and $\mathcal{C}$ denote the sets of exercise items and knowledge concepts, respectively. L_m denotes the length of the interaction sequence $\mathcal{S}_m$ $(m = 1, 2, \cdots, M)$ of the mth student.

Given a new item q_{L_m+1} and its corresponding knowledge concept(s) $\mathcal{C}_{L_m+1}$, the KT model predicts the probability of a correct response as:

$$\mathrm{P}(a_{L_m+1} = 1 \mid \mathcal{S}_m, q_{L_m+1}, \mathcal{C}_{L_m+1}, \mathcal{G}), \tag{2}$$

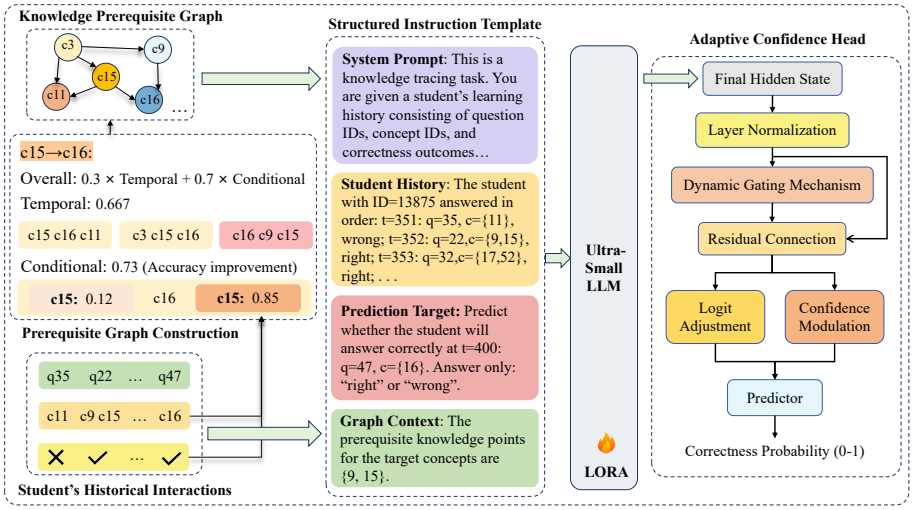

Fig. 1. The overall architecture of GLLM-KT. GLLM-KT first constructs a knowledge prerequisite graph from student learning interaction sequences using Temporal Precedence Score (TPS) and Conditional Dependency Score (CDS). It then formulates each student's interaction sequence into a structured instruction template consisting of System Prompt (SP), Student History (LH), Prediction Target (PT), and Graph Context (GC). This instruction is fed into an ultra-small LLM, whose output representations are processed by an Adaptive Confidence Head (ACH) to produce the final prediction result.

where the knowledge prerequisite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V} = \mathcal{C}$ and an edge set $\mathcal{E}$ that encodes prerequisite relations among nodes in $\mathcal{V}$. The model thus learns a mapping

$$f : (\mathcal{S}_m, q_{L_m+1}, \mathcal{C}_{L_m+1}, \mathcal{G}) \longrightarrow [0, 1], \tag{3}$$

which integrates both temporal learning dynamics and structural dependencies among knowledge concepts.

3.2 GLLM-KT

Knowledge Prerequisite Graph To explicitly model the relationships among knowledge concepts, we construct a directed acyclic knowledge graph from student learning interaction sequences. The graph provides structural priors to help GLLM-KT infer conceptual dependencies among concepts.

Given the interaction sequence of the mth student as $\mathcal{S}_m$ $(m = 1, 2, \cdots, M)$. we infer directed edges among concepts based on two complementary criteria. For a candidate relation $c_{i_1} \rightarrow c_{i_2}$, its Temporal Precedence Score (TPS) reflects how often a student masters knowledge concept c_{i_1} before mastering c_{i_2} as defined

by the following equation:

$$\mathrm{TPS}_{c_{i_1}\to c_{i_2}} = M_{c_{i_1}\to c_{i_2}}/M_{c_{i_2}} \tag{4}$$

where

$$M_{c_{i_1}\to c_{i_2}} = \Big|\big\{m \mid m \in [1,M] \wedge \mathrm{FC}(\mathcal{S}_m, c_{i_1}) < \mathrm{FC}(\mathcal{S}_m, c_{i_2})\big\}\Big| \tag{5}$$

where $\mathrm{FC}(\mathcal{S}_m, c_{i_1})$ and $\mathrm{FC}(\mathcal{S}_m, c_{i_2})$ denote the index position in sequence $\mathcal{S}_m$ of student's first correct response to exercises related to concept c_{i_1} and c_{i_2}, respectively. $M_{c_{i_1}\to c_{i_2}}$ denotes the number of students who masters c_{i_1} before mastering c_{i_2} and $M_{c_{i_2}}$ denotes the total number of students who masters c_{i_2}. The Conditional Dependency Score (CDS) measures how much mastering c_{i_1} helps in learning c_{i_2} as defined by the below equation:

$$\mathrm{CDS}_{c_{i_1}\to c_{i_2}} = \frac{\mathrm{P}_{c_{i_1}\to c_{i_2}} - \mathrm{P}_{c_{i_2}\to c_{i_1}} + 1}{2}, \tag{6}$$

where

$$\mathcal{A}_{m,c_{i_1}\to c_{i_2}} = \big\{n \mid n \in [1,L_m] \wedge c_{i_2} \in \mathcal{C}_n \wedge \mathrm{FC}(\mathcal{S}_m, c_{i_1}) < n\big\}, \tag{7}$$

$$\mathcal{A}_{m,c_{i_2}\to c_{i_1}} = \big\{n \mid n \in [1,L_m] \wedge c_{i_2} \in \mathcal{C}_n \wedge n < \mathrm{FC}(\mathcal{S}_m, c_{i_1})\big\}, \tag{8}$$

$$L_{m,c_{i_1}\to c_{i_2}} = \Big|\big\{n \mid n \in \mathcal{A}_{m,c_{i_1}\to c_{i_2}} \wedge a_n = 1\big\}\Big|, \tag{9}$$

$$L_{m,c_{i_2}\to c_{i_1}} = \Big|\big\{n \mid n \in \mathcal{A}_{m,c_{i_2}\to c_{i_1}} \wedge a_n = 1\big\}\Big|, \tag{10}$$

$$\mathrm{P}_{c_{i_1}\to c_{i_2}} = \frac{\sum_{m=1}^{M} L_{m,c_{i_1}\to c_{i_2}}}{\sum_{m=1}^{M}\Big|\mathcal{A}_{m,c_{i_1}\to c_{i_2}}\Big|}, \quad \mathrm{P}_{c_{i_2}\to c_{i_1}} = \frac{\sum_{m=1}^{M} L_{m,c_{i_2}\to c_{i_1}}}{\sum_{m=1}^{M}\Big|\mathcal{A}_{m,c_{i_2}\to c_{i_1}}\Big|}, \tag{11}$$

where $\mathrm{P}_{c_{i_1}\to c_{i_2}}$ and $\mathrm{P}_{c_{i_2}\to c_{i_1}}$ are the accuracy on concept c_{i_2} after and before mastering concept c_{i_1}, respectively. $\mathcal{A}_{m,c_{i_1}\to c_{i_2}}$ and $\mathcal{A}_{m,c_{i_2}\to c_{i_1}}$ denote sets of all attempts on concept c_{i_2} occurring after and before mastering c_{i_1} in sequence $\mathcal{S}_m$, respectively. $L_{m,c_{i_1}\to c_{i_2}}$ and $L_{m,c_{i_2}\to c_{i_1}}$ represent the number of correct attempts on c_{i_2} after and before mastering c_{i_1} in sequence $\mathcal{S}_m$, respectively. Then, the edge score is defined as:

$$\mathrm{ES}_{c_{i_1}\to c_{i_2}} = \alpha \times \mathrm{TPS}_{c_{i_1}\to c_{i_2}} + (1-\alpha) \times \mathrm{CDS}_{c_{i_1}\to c_{i_2}}, \tag{12}$$

where weight $\alpha \in [0,1]$ balances TPS and CDS. Only edges with scores exceeding a predefined threshold τ are retained in a candidate edge set as following:

$$\mathcal{E} = \big\{(c_{i_1}, c_{i_2}) \mid \mathrm{ES}_{c_{i_1}\to c_{i_2}} > \tau\big\}. \tag{13}$$

To ensure the final knowledge prerequisite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is acyclic, A cycle removal step is applied by iteratively discarding the lowest-scoring edge in any detected directed cycle.

Structured Instruction Template To make KT data suitable for instruction-following LLMs, we formulate each student interaction sequence as instruction–response pair. Each instance consists of four components as below:

$$\mathbf{I} = \langle \mathbf{SP}, \mathbf{SH}, \mathbf{PT}, \mathbf{GC} \rangle, \tag{14}$$

where system prompt **SP** describes the setup and goal of KT task, student history **SH** provides the student's past learning interaction sequence $\mathcal{S}_m$, prediction target **PT** specifies the target exercise q_{L_m+1} to be predicted and its related knowledge concept(s) $\mathcal{C}_{L_m+1}$, requiring a 'right' or 'wrong' response, and graph context **GC** supplies the prerequisite concepts of $\mathcal{C}_{L_m+1}$ retrieved from graph $\mathcal{G}$.

The structured instruction template integrates the student learning behavioral patterns with knowledge relationships under guided task instructions, maintaining a consistent instruction style suitable for LLM fine-tuning. Moreover, the template ensures cross-domain adaptability, as graph context **GC** can be dynamically retrieved from any knowledge graph.

Adaptive Confidence Head The Adaptive Confidence Head (ACH) improves prediction robustness by adaptively filtering feature representations and calibrating output confidence.

Given the final hidden states $\mathbf{H} \in \mathbb{R}^{L_t \times d}$ from an ultra-small LLM, we extract the embedding of the last valid token in the input sequence as the primary representation $\mathbf{h} \in \mathbb{R}^d$ for prediction:

$$\mathbf{h} = \mathbf{H}(l^*, :), \quad l^* = \max \{ l \in [1, L_t] \mid \mathbf{m}_{\text{valid}}(l) = 1 \}. \tag{15}$$

where L_t is the maximum token sequence length that LLM can process, d is the hidden dimension, and $\mathbf{m}_{\text{valid}} \in \{0,1\}^{L_t}$ is a binary mask vector identifying valid token positions with 1 for valid tokens and 0 for padding.

To adaptively emphasize informative features and suppress noise, we employ a dynamic gating mechanism as below:

$$\mathbf{g} = \tanh\{\mathbf{W}_2 \cdot \text{LeakyReLU}\left[\mathbf{W}_1 \cdot \text{LayerNorm}(\mathbf{h}) + \mathbf{b}_1\right] + \mathbf{b}_2\}, \tag{16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d' \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{d'}$, $\mathbf{W}_2 \in \mathbb{R}^{d \times d'}$, $\mathbf{b}_2 \in \mathbb{R}^d$ are learnable parameters and d' is an intermediate dimension. The resulting gating vector $\mathbf{g} \in \mathbb{R}^d$ has values in $(-1, 1)$ and is then integrated with the original features $\mathbf{h}$ by a residual connection with adaptive scaling as below:

$$\mathbf{h}_{\text{res}} = \mathbf{h} + \lambda(\mathbf{h} \odot \mathbf{g}), \tag{17}$$

where $\odot$ denotes the element-wise product and λ is a learnable scalar parameter.

To quantify the inherent uncertainty of each prediction, a confidence score s_c is computed from the enhanced feature $\mathbf{h}_{\text{res}} \in \mathbb{R}^d$ as follows:

$$s_c = \text{Sigmoid}\left[\mathbf{w}_c^\top \cdot \text{LeakyReLU}(\mathbf{W}_c \cdot \mathbf{h}_{\text{res}} + \mathbf{b}_c)\right], \tag{18}$$

where $\mathbf{w}_c \in \mathbb{R}^{d_c}$, $\mathbf{W}_c \in \mathbb{R}^{d_c \times d}$ and $\mathbf{b}_c \in \mathbb{R}^{d_c}$ are learnable parameters. d_c is the hidden dimension of the confidence estimation branch. The output $s_c \in [0, 1]$ represents the model's confidence in its prediction.

The final prediction $\hat{y}_n$ is obtained by integrating the confidence score s_c and a temperature-scaled logit as below:

$$\hat{y}_n = \text{Sigmoid}\left(\frac{\mathbf{w}_z^\top \cdot \mathbf{h}_{\text{res}} + b_z}{\sigma} \cdot s_c\right), \tag{19}$$

where $\mathbf{w}_z \in \mathbb{R}^d$ and $b_z \in \mathbb{R}$ are learnable parameters. σ is the temperature parameter.

The loss function combines binary cross-entropy with focal weighting as below:

$$\mathcal{L} = -\sum_n \omega_n \left[a_n \log \hat{y}_n + (1 - a_n) \log(1 - \hat{y}_n) \right], \tag{20}$$

where ω_n and a_n denote the balancing parameter and the actual response correctness, respectively. This formulation enhances both prediction accuracy and model calibration for KT.

4 Experiments

In this section, we present comprehensive experimental results to validate the effectiveness of GLLM-KT, including comparison experiments, ablation study, hyperparameter sensitivity analysis and case study.

4.1 Experimental Setup

Datasets All experiments are conducted on four benchmark datasets in KT, i.e. ASSIST09[3], ASSIST12[4], EdNet-KT1[5] and Junyi[6]. Table 1 summarizes the detailed statistics of each dataset, including the number of students, learning interactions, exercise items, knowledge concepts and the average learning sequence length $\bar{L}_m$. Following a common practice, each dataset is split into training (70%), validation (10%) and test (20%) sets.

Table 1. Statistics of benchmark datasets

Dataset	#Student	#Interaction	#Concept	#Item	$\bar{L}_m$
ASSIST09	3,628	273,381	101	16,867	75.4
ASSIST12	25,266	2,621,308	198	50,918	103.7
EdNet-KT1	781,213	724,437,429	188	12,252	979.2
Junyi	68,055	16,206,970	1326	25,784	238.1

[3] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data
[4] https://sites.google.com/site/assistmentsdata/2012-13-school-data-with-affect
[5] https://github.com/riiid/ednet?tab=readme-ov-file
[6] https://base.ustc.edu.cn/data/tests/junyi/

Baseline Methods We compare GLLM-KT with four deep learning-based methods, including DKT, AKT, GKT, and MGKT, as well as two LLM-based prompting methods, namely GPT-4.1 [18, 19] and DeepSeek-v3.1 [20, 21]. All baseline methods are implemented using their official configurations. For GPT-4.1 and DeepSeek-v3.1, the APIs only return discrete prediction outputs rather than calibrated probability scores; therefore, AUC values cannot be computed and are reported as "–".

Implementation Details GLLM-KT is based on the Qwen3 family [10] with parameter scales of 0.6 billion (Qwen3-0.6B), 1.7 billion (Qwen3-1.7B) and 4 billion (Qwen3-4B). Each model is fine-tuned with LoRA using a rank $r = 16$ and a scaling factor $\alpha = 32$ applied to all attention matrices. GLLM-KT is trained with the AdamW optimizer using a learning rate of 2×10^{-5} and a weight decay of 0.01, a cosine learning rate schedule with 500 warm-up steps, a batch size of 64, 5 epochs, and a sequence length $n = 50$. The hidden dimensions for GLLM-KT models based on Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B are set to 1024, 2048, and 2560, respectively.

The knowledge prerequisite graph is constructed with threshold $\tau = 0.6$ and temporal/conditional weight $\alpha = 0.3$. The ACH module incorporates a hidden layer with dimension $d' = 256$, dropout set to 0.1, and LeakyReLU activation with a negative slope of 0.01. The temperature parameter σ is initialized to 1, and the balancing parameter ω_n is set according to the ratio of positive and negative samples in each dataset. The confidence branch is implemented as a two-layer perceptron with dimension $d_c = 128$.

All experiments are performed on NVIDIA RTX 4090 GPUs.

Evaluation Metrics All methods are evaluated using six standard metrics, i.e. Area Under the Curve (AUC), Accuracy (ACC), F1 score (F1), Precision (PRE), Recall (REC) and Root Mean Square Error (RMSE).

4.2 Comparison Experiments

Table 2 presents the performance comparison on four benchmark datasets. As can be shown, MGKT outperforms other methods only on EdNet-KT1 dataset indicates the advantage of MGKT in handling extremely large and heterogeneous learning traces. The underperformance of GPT-4.1 and DeepSeek-v3.1 on all metrics except PRE is primarily due to their lack of domain-specific adaptation and structured knowledge priors. In contrast, GLLM-KT-4B achieves the best or second-best performance to all baseline methods in AUC, ACC, F1 and RMSE across all datasets. Even GLLM-KT-0.6B attains comparable results to several deep-learning methods like AKT and GKT. The result indicates that the instruction-based fine-tuning and graph context **GC** help maintain performance under constrained parameter budgets. Furthermore, there is a clear trend of performance improvement as the parameter size increases from 0.6 billion to 4 billion, which shows consistent scalability and robustness across parameter sizes.

Table 2. Performance comparisons on benchmark datasets. Best and second-best results are in **bold** and underlined, respectively. ↑/↓ indicates that higher/lower is better.

Dataset	Metric	DKT	AKT	GKT	MGKT	GPT 4.1	DeepSeek v3.1	GLLM-KT 0.6B	GLLM-KT 1.7B	GLLM-KT 4B
ASSIST09	AUC↑	0.746	0.715	0.730	0.739	–	–	0.716	0.747	**0.752**
	ACC↑	0.703	0.685	0.698	0.697	0.687	0.611	0.697	0.706	**0.711**
	F1↑	0.779	0.770	0.776	0.768	0.754	0.625	0.773	0.782	**0.788**
	PRE↑	0.713	0.693	0.709	0.740	0.807	**0.852**	0.695	0.717	0.725
	REC↑	0.859	**0.867**	0.858	0.798	0.708	0.494	0.871	0.827	0.830
	RMSE↓	0.440	0.451	0.446	0.446	0.560	0.624	0.449	0.436	**0.434**
ASSIST12	AUC↑	0.680	0.653	**0.696**	0.656	–	–	0.667	0.682	0.693
	ACC↑	0.699	0.691	0.700	0.691	0.652	0.563	0.696	0.697	**0.704**
	F1↑	0.809	0.806	0.809	0.792	0.742	0.600	0.808	0.811	**0.812**
	PRE↑	0.714	0.705	0.715	0.736	0.741	**0.783**	0.711	0.724	0.727
	REC↑	0.933	**0.942**	0.931	0.857	0.743	0.486	0.935	0.922	0.920
	RMSE↓	0.445	0.450	0.442	0.455	0.590	0.661	0.447	0.443	**0.438**
EdNet-KT1	AUC↑	0.646	0.628	0.652	**0.724**	–	–	0.633	0.662	0.679
	ACC↑	0.635	0.627	0.633	**0.685**	0.586	0.484	0.637	0.649	0.654
	F1↑	0.739	0.732	0.715	**0.759**	0.654	0.382	0.733	0.739	0.746
	PRE↑	0.650	0.647	0.672	0.724	0.682	**0.728**	0.644	0.683	0.691
	REC↑	**0.857**	0.843	0.764	0.797	0.628	0.259	0.851	0.805	0.811
	RMSE↓	0.474	0.477	0.473	**0.450**	0.644	0.719	0.471	0.465	0.461
Junyi	AUC↑	0.672	0.712	0.680	0.739	–	–	0.702	0.736	**0.747**
	ACC↑	0.730	0.741	0.728	0.741	0.706	0.690	0.733	0.739	**0.750**
	F1↑	0.834	0.837	0.833	0.827	0.793	0.766	0.830	0.847	**0.851**
	PRE↑	0.749	0.770	0.750	0.788	0.795	**0.841**	0.768	0.773	0.785
	REC↑	**0.941**	0.916	0.936	0.870	0.790	0.703	0.903	0.937	0.929
	RMSE↓	0.431	0.421	0.429	0.425	0.542	0.556	0.433	0.424	**0.415**

4.3 Ablation Study

To evaluate the contribution of each component in GLLM-KT, we conduct ablation experiments on the ASSIST09 and EdNet-KT1 datasets. As shown in Table 3, we compare the quantitative results of the full GLLM-KT model against three variants, each with one core module removed, i.e. knowledge prerequisite graph, ACH or LoRA fine-tuning.

As can be seen, the full GLLM-KT model achieves the best performance on both datasets. The removal of ACH leads to a consistent performance drop of about 0.01 in AUC, ACC and F1. Removing the knowledge prerequisite graph leads to 0.012∼0.025 decline in AUC, ACC and F1. Especially, the most dramatic performance decline is observed when the LoRA fine-tuning strategy is ablated, with a decrease in AUC by 0.062/0.045, ACC by 0.022/0.032, F1 by 0.056/0.041 and an increase in RMSE by 0.036/0.059 on two datasets, respectively. The results validate the effectiveness of all three components for the model's superior performance.

4.4 Hyperparameter Sensitivity Analysis

We further analyze the sensitivity of GLLM-KT with respect to three key hyperparameters, including the LoRA rank r, learning rate and graph edge thresh-

Table 3. Ablation study results of GLLM-KT.

Model	ASSIST09				EdNet-KT1			
	AUC↑	ACC↑	F1↑	RMSE↓	AUC↑	ACC↑	F1↑	RMSE↓
GLLM-KT	**0.752**	**0.711**	**0.788**	**0.434**	**0.679**	**0.654**	**0.746**	**0.461**
w/o ACH	0.742	0.705	0.775	0.437	0.666	0.643	0.736	0.472
w/o Knowledge Graph	0.731	0.694	0.770	0.439	0.654	0.642	0.732	0.472
w/o LoRA fine-tuning	0.690	0.689	0.732	0.470	0.634	0.622	0.705	0.520

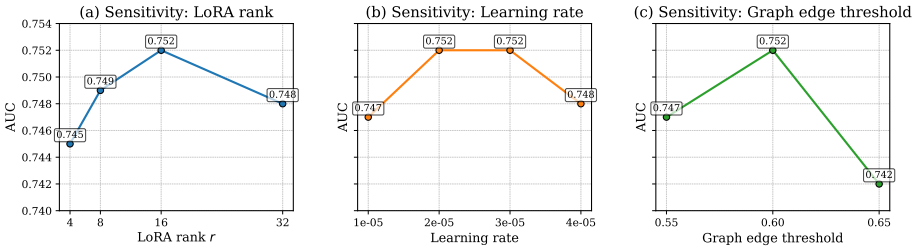

Fig. 2. Hyperparameter sensitivity analysis results on ASSIST09 according to (a) LoRA rank, (b) learning rate and (c) graph edge threshold τ.

old τ. The evaluated values include LoRA ranks $r \in \{4, 8, 16, 32\}$, learning rates in $\{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}, 4 \times 10^{-5}\}$, and graph edge thresholds in $\{0.55, 0.60, 0.65\}$. Fig. 2 illustrates their influence on AUC using the ASSIST09 dataset. As can be observed, GLLM-KT performs the best when the LoRA rank, learning rate and graph edge threshold are set to 16, 2×10^{-5} and 0.60, respectively. In addition, deviations from the optimal settings definitely result in different degrees of performance degradation. For example, a lower LoRA rank may limits model's representational ability, while a higher learning rate can lead to training instability.

4.5 Case Study

To illustrate how GLLM-KT performs reasoning over structured learning evidence, we present a representative case from the Junyi dataset in Fig. 3. The case shows that student has mastered the prerequisite knowledge concepts c_6, c_{14}, c_{16} before attempting an exercise item related to the target concept c_{29}. The structural dependencies enable GLLM-KT to estimate a high correctness probability of 0.89 for the student's response to the item, consistent with the student's actual performance. The case indicates that GLLM-KT can transfer knowledge state information through the graph structure and thus produce interpretable and reliable predictions.

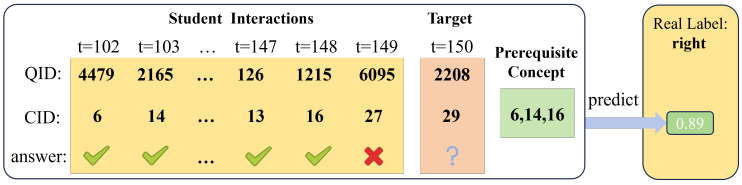

Fig. 3. A case study of GLLM-KT in student performance prediction

5 Conclusion

In this paper, we proposed GLLM-KT, a graph-incorporated KT model based on ultra-small LLM. It consists of three key components, including a knowledge prerequisite graph to capture conceptual dependencies, a structured instruction template to adapt KT data for LLM processing and an ACH to improve prediction stability and interpretability. Extensive experiments on four benchmark datasets have shown that GLLM-KT outperforms several state-of-the-art deep learning and LLM-based prompting methods in both prediction accuracy and generalization ability.

In the future, we plan to extend the proposed framework to cross-domain settings, incorporate cognitive features such as learning pace and forgetting curves, and develop student profile modeling based on reinforcement learning to further enhance the personalization and interpretability of KT systems.

Acknowledgments. This work was funded in part by the National Natural Science Foundation of China under Grant 62377010 and in part by the Natural Science Foundation of Hunan Province, China, under Grant 2024JJ6721.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction **4**(4), 253–278 (1994)
2. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems **28**, 505–513 (2015)
3. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web (WWW), pp. 765–774. International World Wide Web Conferences Steering Committee, Perth (2017)
4. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. In: Proceedings of the 12th International Conference on Educational Data Mining (EDM), pp. 384–389. International Educational Data Mining Society, Montreal (2019)

5. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 2330–2339. ACM, Virtual Event (2020)
6. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 156–163. IEEE, Thessaloniki (2019)
7. Kang, R., Li, X., Liu, G., Wang, L.: Multiple graph knowledge tracing based on LSTM-attention hypergraph convolution and forgetting effect. IEEE Transactions on Computational Social Systems (2025)
8. Neshaei, S.P., Davis, R.L., Hazimeh, A., Lazarevski, B., Dillenbourg, P., Käser, T.: Towards modeling learner performance with large language models. In: Proceedings of the International Conference on Educational Data Mining, pp. 759–768 (2024)
9. Li, R., Wu, S., Wang, J., Zhang, W.: CIKT: A collaborative and iterative knowledge tracing framework with large language models. arXiv preprint arXiv:2505.17705 (2025)
10. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (ICLR), pp. 1–3 (2022)
12. Feng, M., Heffernan, N., Koedinger, K.: ASSISTments 2009–2010 Skill Builder Dataset. ASSISTments Foundation (2009). Available at: https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010
13. Feng, M., Heffernan, N., Koedinger, K.: ASSISTments 2012–2013 School Data with Affect. ASSISTments Foundation (2012). Available at: https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect
14. Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., Heo, J.: Ednet: A large-scale hierarchical dataset in education. In: International Conference on Artificial Intelligence in Education, pp. 69–73. Springer (2020)
15. Chang, H.-S., Hsu, H.-J., Chen, K.-T.: Modeling Exercise Relationships in E-Learning: A Unified Approach. In: International Conference on Educational Data Mining (EDM), 2015
16. Salomons, N., Scassellati, B.: Time-dependant Bayesian knowledge tracing—Robots that model user skills over time. Frontiers in Robotics and AI, **10**, 1249241 (2024)
17. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance Factors Analysis—A New Alternative to Knowledge Tracing. Online Submission. ERIC (2009)
18. OpenAI. Introducing GPT-4.1 in the API. OpenAI blog, April 14, 2025. Available: https://openai.com/index/gpt-4-1/
19. OpenAI. GPT-4.1 model — API documentation. OpenAI Developer Platform, 2025. Available: https://platform.openai.com/docs/models/gpt-4.1
20. A. Liu et al. DeepSeek-V3 Technical Report. arXiv preprint arXiv:2412.19437 (2024).
21. DeepSeek-AI. DeepSeek-V3.1 Release Notes. DeepSeek official news / API docs (2025). Available: https://api-docs.deepseek.com/news/news250821