CIKT: A Collaborative and Iterative Knowledge Tracing Framework with Large Language Models

Runze Li[†], **Siyu Wu**[†], **Jun Wang**[†*], **Wei Zhang**[†‡*]
[†]School of Computer Science and Technology, East China Normal University
[‡]Shanghai Innovation Institute
51265901021@stu.ecnu.edu.cn, 52265901039@stu.ecnu.edu.cn
wongjun@gmail.com, zhangwei.thu2011@gmail.com

Abstract

Knowledge Tracing (KT) aims to model a student's learning state over time and predict their future performance. However, traditional KT methods often face challenges in explainability, scalability, and effective modeling of complex knowledge dependencies. While Large Language Models (LLMs) present new avenues for KT, their direct application often struggles with generating structured, explainable student representations and lacks mechanisms for continuous, task-specific refinement. To address these gaps, we propose Collaborative Iterative Knowledge Tracing (CIKT), a framework that harnesses LLMs to enhance both prediction accuracy and explainability. CIKT employs a dual-component architecture: an Analyst generates dynamic, explainable user profiles from student historical responses, and a Predictor utilizes these profiles to forecast future performance. The core of CIKT is a synergistic optimization loop. In this loop, the Analyst is iteratively refined based on the predictive accuracy of the Predictor, which conditions on the generated profiles, and the Predictor is subsequently retrained using these enhanced profiles. Evaluated on multiple educational datasets, CIKT demonstrates significant improvements in prediction accuracy, offers enhanced explainability through its dynamically updated user profiles, and exhibits improved scalability. Our work presents a robust and explainable solution for advancing knowledge tracing systems, effectively bridging the gap between predictive performance and model transparency.

1 Introduction

Knowledge Tracing (KT) (Corbett and Anderson, 1994) is a foundational task in educational data mining and intelligent tutoring systems, aiming to model a student's evolving knowledge state from their historical learning interactions to accurately predict future performance, thereby facilitating personalized learning and targeted interventions. While early approaches like Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994) and its extensions (Pardos and Heffernan, 2011, 2010) offered interpretable parameters, they often struggled with the complex temporal dependencies of learning processes. Deep learning-based KT (DLKT) models subsequently emerged, significantly advancing predictive accuracy. Pioneering models such as Deep Knowledge Tracing (DKT) (Piech et al., 2015) with Recurrent Neural Networks, and memory-augmented architectures like DKVMN (Zhang et al., 2017), laid crucial groundwork (Liu et al., 2019; Nagatani et al., 2019). Further advancements, including Transformer-based models like SAKT (Pandey and Karypis, 2019), AKT (Ghosh et al., 2020), and LPKT (Shen et al., 2021) (which incorporated cognitive dynamics), alongside innovations like FoLiBi's linear forgetting mechanisms (Im et al., 2023) and the integration of side information (Wang et al., 2021; Pandey and Srivastava, 2020) or graph structures (Nakagawa et al., 2019; Yang et al., 2025; Wang et al., 2025), have continued to push KT performance boundaries.

Despite these significant strides in predictive power, achieving robust explainability remains a persistent challenge in the DLKT landscape (Minn et al., 2022; Zhao et al., 2020). Although various strategies have been explored—from inherently interpretable components (Zhang et al., 2017; Shen et al., 2021) and post-hoc analyses (Scruggs et al., 2019) to aligning models with learning theories (Chen et al., 2023; Cui et al., 2024)—many DLKT models remain substantially opaque. This lack of transparency can hinder their adoption and trustworthiness in high-stakes educational settings where understanding the model's reasoning is crucial.

The transformative capabilities of LLMs, demonstrated across specialized domains like scien-

[*]Corresponding authors

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 19321–19334
November 4-9, 2025 ©2025 Association for Computational Linguistics

tific discovery (Pyzer-Knapp et al., 2022; Merchant et al., 2023) and automated research assistance (Wang et al., 2024b,a; Lu et al., 2024; Huang et al., 2023; Tyser et al., 2024), offer promising new avenues for addressing KT's dual challenges. However, directly applying general-purpose LLMs to the nuanced task of knowledge tracing introduces distinct difficulties: (1) eliciting structured, interpretable representations of dynamic student knowledge states beyond mere task-specific predictions; (2) optimizing LLM behavior for KT without abundant, explicit preference signals or fine-grained supervision for explainability; and (3) resolving the inherent tension between maximizing predictive accuracy and maintaining KT process explainability. Moreover, many current LLM applications operate statically post-deployment, lacking mechanisms for continuous self-improvement based on domain-specific feedback.

To address these multifaceted limitations, we propose Collaborative Iterative Knowledge Tracing (CIKT), a framework architected around two core LLM-based components: an Analyst that generates structured, interpretable student profiles from historical responses, and a Predictor that leverages these profiles for future performance forecasting. The cornerstone of CIKT is an iterative learning strategy employing Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024). This mechanism facilitates reciprocal enhancement: the Predictor's accuracy, conditioned on Analyst-generated profiles, provides reinforcement-style feedback to progressively refine the Analyst. Subsequently, the Predictor is retrained with these enhanced profiles, completing a collaborative optimization loop. Both the Analyst and Predictor are built upon a large-scale pre-trained language model backbone, ensuring flexibility and powerful representation learning.

The major contributions of this paper are summarized as follows:

- We propose a **collaborative knowledge tracing framework** that explicitly models student knowledge states via an Analyst and utilizes these dynamic profiles for predictive tasks through a Predictor.
- We introduce **an iterative optimization strategy** based on reinforcement-style feedback, enabling mutual refinement between the Analyst and Predictor to improve both the quality of generated profiles and overall predictive performance.
- We conduct extensive experiments on multiple educational datasets, demonstrating that our CIKT framework outperforms existing KT models in predictive accuracy while simultaneously offering enhanced explainability through its generated student profiles.

2 Related Work

Knowledge tracing (KT) (Corbett and Anderson, 1994), a key task in educational data mining, models students' evolving knowledge states to predict future performance. Early models like Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994) used binary mastery variables and explainable parameters for learning/forgetting dynamics. Extensions to BKT (Pardos and Heffernan, 2011, 2010) and other machine learning methods (Pavlik et al., 2009) aimed to improve accuracy and flexibility. However, these models struggled with complex temporal dependencies and latent interactions.

Deep learning-based knowledge tracing (DLKT) models emerged to overcome these limitations. Deep Knowledge Tracing (DKT) (Piech et al., 2015) notably used RNNs to learn latent representations from student interactions. DKVMN (Zhang et al., 2017) later enhanced the structure by using key-value memory networks for concept mastery tracking. These pioneering DLKT models established a foundation for later work (Liu et al., 2019; Nagatani et al., 2019; Shen et al., 2021).

Recent advances leverage attention mechanisms and Transformers to better model long-range dependencies. Models like SAKT (Pandey and Karypis, 2019), AKT (Ghosh et al., 2020), LPKT (Shen et al., 2021) (with memory-aligned gates), and FoLiBi (Im et al., 2023) (with linear forgetting) improved accuracy and explainability by modeling contextual and cognitive dynamics. Integrating side information like temporal or contextual features (Wang et al., 2021; Pandey and Srivastava, 2020) also enhanced KT performance. Graph-based methods (Nakagawa et al., 2019; Yang et al., 2025; Wang et al., 2025) model concept and interaction dependencies for better knowledge representation.

Explainability remains a key DLKT concern despite these developments. Efforts include inherently explainable architectures (Zhang et al., 2017; Shen et al., 2021; Minn et al., 2022), post-hoc analysis of trained models (e.g., attention weights) (Zhao et al., 2020; Scruggs et al., 2019), and integrated modules like attention or cognitive mechanisms. However, these methods often face generalizability issues and task-specific design dependencies. Other works target explainability by aligning models with learning theories (Chen et al., 2023; Cui et al., 2024). Still, most DLKT models remain opaque, posing audit challenges in high-stakes education.

The adaptation of Large Language Models (LLMs) for specialized applications in various vertical domains shows considerable promise beyond general-purpose tasks. For instance, their capabilities are harnessed for nuanced information processing, such as automating scientific literature retrieval (Wang et al., 2024b), generating domain-specific survey papers (Wang et al., 2024a), aiding complex data analysis in scientific discovery (e.g., material discovery (Pyzer-Knapp et al., 2022; Merchant et al., 2023)), supporting prompt-driven research pipelines (Lu et al., 2024), evaluating specialized content like scientific papers (Tyser et al., 2024), and assisting in domain-specific coding solutions (Huang et al., 2023). While these domain-specific adaptations often achieve notable performance through fine-tuning or sophisticated prompting, a common limitation is their static operation post-deployment; they typically lack embedded processes for continuous self-iteration and performance enhancement based on ongoing, domain-specific feedback. Addressing this crucial gap, our work proposes a novel collaborative iterative optimization framework specifically designed to empower LLMs to continuously refine their own effectiveness for the specialized task at hand.

3 Methodology

3.1 Overall Framework

To enhance both the accuracy and explainability of KT, we propose CIKT, a framework leveraging the capabilities of LLMs. CIKT revolves around two core intelligent components: an Analyst responsible for generating rich, structured student profiles from historical interaction data, and a Predictor that utilizes these profiles, alongside interaction history, to forecast student performance. The synergy between the Analyst and Predictor is cultivated through a meticulously designed four-stage iterative process, illustrated in Figure 1. This cycle begins with **Distillation**, where the Analyst learns foundational profiling capabilities from curated annotations initially provided by a large teacher model. Next, in the **Profiling** stage, the trained Analyst generates comprehensive user profiles for student data. Subsequently, during **Reasoning**, the Predictor is trained to predict outcomes using these profiles and historical interactions. Finally, the **Iteration** stage employs a refinement loop where feedback from the Predictor's performance, guided by Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024) principles, is used to further optimize the Analyst via reinforcement learning. This iterative process facilitates mutual improvement, enhancing both the quality of the generated profiles and the accuracy of predictions.

3.2 Stage 1: Distillation

This initial Distillation stage aims to endow the Analyst with the foundational capability to generate structured and informative user profiles from raw student interaction data. The process begins by leveraging a large-scale teacher model, LLM$_{\text{teacher}}$ (e.g., GPT-4o (Hurst et al., 2024)), to process historical interaction sequences from a subset of students. For a student s, their sequence is denoted as $\mathcal{S}_s = \{(e_1, r_1), (e_2, r_2), \ldots, (e_N, r_N)\}$, where e_i is the i-th exercise and $r_i \in \{0, 1\}$ its binary correctness. The LLM$_{\text{teacher}}$ produces initial textual profiles:

$$\mathbf{p}_{s,\text{teacher}} = \text{LLM}_{\text{teacher}}(\mathcal{S}_s) \tag{1}$$

These profiles, $\mathbf{p}_{s,\text{teacher}}$, are designed as textual outputs capturing the student's knowledge state, including aspects like mastery levels across knowledge concepts, inferred learning patterns, and potential difficulties. Subsequently, these profiles undergo a manual curation process where their format and content are reviewed and corrected; only selected, high-quality profiles, denoted as $\mathbf{p}^*_{s,\text{teacher}}$, that accurately reflect student understanding are retained for training.

The Analyst, parameterized by θ_A and based on our chosen backbone LLM architecture, is then fine-tuned via supervised learning using pairs of student sequences ($\mathcal{S}_s$) and their corresponding curated teacher profiles ($\mathbf{p}^*_{s,\text{teacher}}$) from a training set $\mathcal{D}_{\text{train}}$. The Analyst learns to map an input sequence $\mathcal{S}_s$ to its own profile generation $\mathbf{p}_{s,\text{analyst}}$:

$$\mathbf{p}_{s,\text{analyst}} = \text{Analyst}(\mathcal{S}_s; \theta_A) \tag{2}$$

The training objective is to minimize a distillation loss, $\mathcal{L}_{\text{Distill}}$. Given that the profiles are textual, this loss is formulated as a token-level cross-entropy ($\mathcal{L}_{\text{CE}}$) between the Analyst-generated profiles and

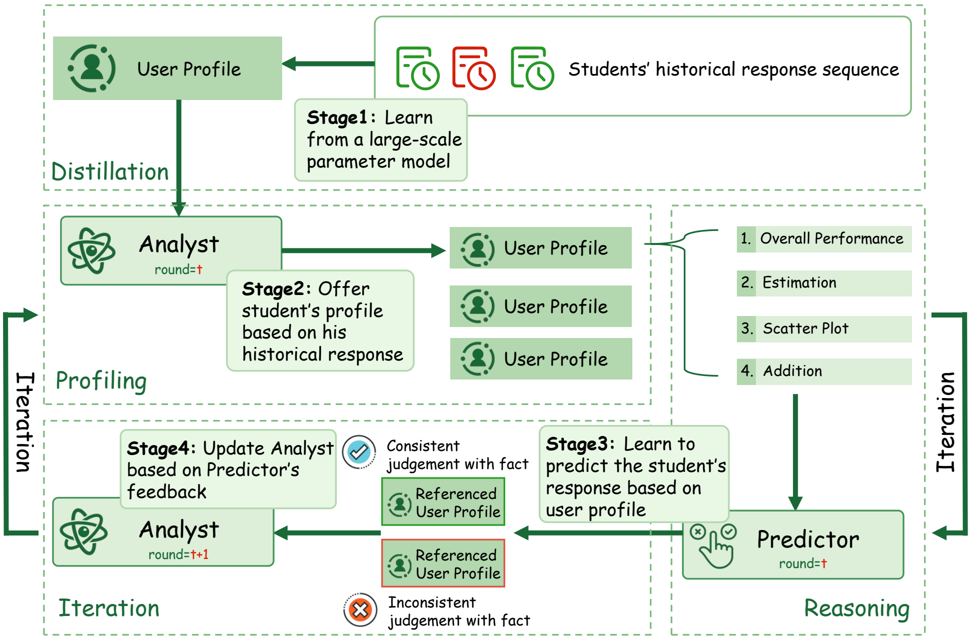
User Profile
Students' historical response sequence
Stage1: Learn from a large-scale parameter model
Distillation
Analyst
round=t
User Profile
User Profile
User Profile
Stage2: Offer student's profile based on his historical response
Profiling
1. Overall Performance
2. Estimation
3. Scatter Plot
4. Addition
Iteration
Iteration
Stage4: Update Analyst based on Predictor's feedback
Consistent judgement with fact
Referenced User Profile
Stage3: Learn to predict the student's response based on user profile
Analyst
round=t+1
Referenced User Profile
Predictor
round=t
Iteration
Inconsistent judgement with fact
Reasoning

Figure 1: **The CIKT framework**, illustrating the collaborative four-stage process involving the Analyst and Predictor: distillation, profiling, reasoning, and iteration.

the curated teacher profiles:

$$\mathcal{L}_{\text{Distill}}(\theta_A) = \sum_{s \in \mathcal{D}_{\text{train}}} \mathcal{L}_{\text{CE}}(\mathbf{p}^*_{s,\text{teacher}}, \mathbf{p}_{s,\text{analyst}}) \tag{3}$$

This supervised distillation phase equips the Analyst with a robust initial model for generating meaningful user profiles.

3.3 Stage 2: Profiling

Following the initial foundation building in the Distillation stage, the Analyst, now equipped with its learned parameters θ_A, is applied to generate user profiles for all relevant student data. The primary objective of this Profiling stage is to transform raw student historical interaction sequences into rich, structured profile representations that will inform the subsequent prediction tasks.

For each student s with a historical interaction sequence $\mathcal{S}_s = \{(e_1, r_1), (e_2, r_2), \dots, (e_N, r_N)\}$, the trained Analyst synthesizes a corresponding user profile $\mathbf{p}_s$. This process can be represented as:

$$\mathbf{p}_s = \text{Analyst}(\mathcal{S}_s; \theta_A) \tag{4}$$

where $\mathbf{p}_s$ is the textual profile generated by the Analyst based on the student's historical interactions. This profiling step is systematically applied across the entire dataset, including the training, validation, and test sets. The resulting set of user profiles $\{\mathbf{p}_s\}$ serves as a crucial augmented input, alongside the original interaction sequences $\{\mathcal{S}_s\}$, for training and evaluating the Predictor in the subsequent Reasoning stage (Section 3.4). The quality and informativeness of these profiles are paramount for the Predictor's ability to make accurate and nuanced performance forecasts.

3.4 Stage 3: Reasoning

The "Reasoning" stage centers on training the Predictor, parameterized by θ_P. Its objective is to accurately forecast a student's performance $y_{s,t}$ (where $y_{s,t} \in \{0, 1\}$ indicates binary correctness) on a subsequent learning exercise e_t. To achieve this, the Predictor utilizes a combination of the student's historical interaction sequence $\mathcal{H}_{s,t-1} = \{(e_1, r_1), \dots, (e_{t-1}, r_{t-1})\}$ (where e_i is an exercise and r_i its correctness), the corresponding user profile $\mathbf{p}_{s,t-1}$ generated by the Analyst (i.e., $\mathbf{p}_{s,t-1} = \text{Analyst}(\mathcal{H}_{s,t-1}; \theta_A)$), and information pertaining to the target exercise e_t. The Predictor then outputs the predicted probability of a correct response:

$$\hat{y}_{s,t} = \text{Predictor}(\mathcal{H}_{s,t-1}, \mathbf{p}_{s,t-1}, e_t; \theta_P) \tag{5}$$

Training of the Predictor is conducted via supervised fine-tuning. Given a training set $\mathcal{D}_{\text{train}}$ comprising instances of $(\mathcal{H}_{s,t-1}, \mathbf{p}_{s,t-1}, e_t, y_{s,t})$, the parameters θ_P are optimized by minimizing the binary cross-entropy loss function, $\mathcal{L}_{\text{Predict}}$. This loss encourages the predicted probabilities $\hat{y}_{s,t}$ to

Dataset	**ASSIST2009**			**ASSIST2012**			**Eedi**			**BePKT**		
Metrics	ACC	ACC$_{len>15}$	F1	ACC	ACC$_{len>15}$	F1	ACC	ACC$_{len>15}$	F1	ACC	ACC$_{len>15}$	F1
DKT	0.737	0.744	0.808	0.745	0.737	0.828	0.712	0.707	0.814	0.685	0.680	0.565
AKT	0.725	0.716	0.801	0.744	0.725	0.821	0.719	0.702	0.807	0.684	0.665	0.559
SAKT	0.743	0.755	0.810	0.743	0.747	0.830	0.722	0.725	0.816	0.690	0.701	0.571
LPKT	0.738	0.750	0.812	0.731	0.751	$\underline{0.832}$	0.728	$\underline{0.729}$	0.818	0.681	0.679	0.573
IKT	0.726	0.710	0.802	0.745	0.741	0.807	0.719	0.701	0.793	0.673	0.688	0.561
DIMKT	$\underline{0.749}$	0.748	$\underline{0.810}$	$\underline{0.747}$	$\underline{0.756}$	0.816	$\underline{0.733}$	0.727	$\underline{0.819}$	0.703	0.701	0.569
DKVMN	0.724	0.729	0.793	0.735	0.727	0.760	0.719	0.715	0.766	0.684	0.697	0.573
GPT-4o	0.732	$\underline{0.756}$	0.794	0.720	0.619	0.802	-	-	-	0.609	0.616	0.541
Deepseek-R1	0.669	0.665	0.762	0.646	0.671	0.770	-	-	-	$\underline{0.718}$	$\underline{0.728}$	$\underline{0.584}$
CIKT-Llama3.1-8B	0.775	**0.781**	**0.827**	0.774	0.784	0.847	0.770	0.775	0.834	0.749	0.757	0.657
CIKT-Qwen2.5-7B	**0.777**	0.778	0.820	**0.780**	**0.790**	**0.852**	**0.777**	**0.781**	**0.836**	**0.762**	**0.773**	**0.660**
improv.	+3.74%	+4.13%	+2.10%	+4.42%	+4.50%	+2.40%	+6.00%	+7.13%	+2.08%	+6.12%	+6.18%	+13.01%

Table 1: Results of the main experiments.

closely align with the true outcomes $y_{s,t}$:

$$\mathcal{L}_{\text{Predict}}(\theta_P) = -\sum_{(s,t)\in\mathcal{D}_{\text{train}}}\Big[y_{s,t}\log(\hat{y}_{s,t}) + (1-y_{s,t})\log(1-\hat{y}_{s,t})\Big] \tag{6}$$

This process enables the Predictor to learn complex relationships between past learning activities, the summarized knowledge state encapsulated in the profile, and future performance, thereby effectively reasoning to arrive at its predictions.

Dataset	ASSIST2009	ASSIST2012	Eedi	BePKT
# Responses	0.4m	2.7m	17.8m	23.9k
# Sequences	8.3k	67.1k	475.4k	1.9k
# Questions	6.9k	53.1k	2.7k	0.5k
# Concepts	200	265	386	100

Table 2: Statistics of the preprocessed datasets.

3.5 Stage 4: Iteration

The Iteration stage is pivotal to our CIKT framework's capacity for progressive enhancement of user profile quality and, consequently, knowledge tracing prediction accuracy. This stage implements an iterative refinement loop where the Analyst is optimized using feedback from the Predictor's performance. This optimization is guided by Kahneman-Tversky Optimization (KTO) principles (Ethayarajh et al., 2024), which leverage binary feedback indicating whether a generated profile contributes to an accurate prediction by the Predictor.

The iterative cycle unfolds as follows:

1. **Profile Generation:** The current Analyst, parameterized by θ_A and denoted as a policy π_{θ_A}, generates a user profile $\mathbf{p}_t$ from a given student's historical interaction sequence $\mathbf{x}_t$:
$$\mathbf{p}_t \sim \pi_{\theta_A}(\cdot \mid \mathbf{x}_t) \tag{7}$$
2. **Prediction and Reward Computation:** The current Predictor (parameterized by θ_P, denoted f_{θ_P}) utilizes $\mathbf{p}_t$, $\mathbf{x}_t$, and potentially the next exercise e_{t+1}, to predict student performance $\hat{y}_{t+1}$. This prediction is compared against the ground truth y_{t+1} to yield a binary reward r_{t+1}:
$$\hat{y}_{t+1} = f_{\theta_P}(\mathbf{x}_t, \mathbf{p}_t, e_{t+1}) \tag{8}$$
$$r_{t+1} = \begin{cases} +1, & \text{if } \hat{y}_{t+1} = y_{t+1} \\ -1, & \text{if } \hat{y}_{t+1} \neq y_{t+1} \end{cases} \tag{9}$$
3. **Analyst Optimization:** The reward r_{t+1} guides the update of the Analyst's parameters θ_A, encouraging the generation of profiles that lead to accurate Predictor outcomes. The KTO loss function for a batch of instances is:
$$\mathcal{L}_{\text{KTO}}(\theta_A) = -\sum_t \left[r_{t+1}\cdot\log\pi_{\theta_A}(\mathbf{p}_t \mid \mathbf{x}_t)\right] \tag{10}$$
where the sum is over instances t in a training batch. This update resembles a policy gradient step, with $\log\pi_{\theta_A}(\mathbf{p}_t \mid \mathbf{x}_t)$ being the log-probability of generating profile $\mathbf{p}_t$.
4. **Predictor Re-training:** After the Analyst is updated (to $\theta_A^{\text{updated}}$) and its profiling capabilities are enhanced, the Predictor can be retrained or further fine-tuned. This uses profiles $\mathbf{p}^{\text{new}}$ generated by the improved Analyst and minimizes the predictive loss (Equation 6):
$$\theta_P \leftarrow \arg\min_{\theta_P} \sum_{(s,t)\in\mathcal{D}_{\text{train}}} \mathcal{L}_{\text{CE}}\big(f_{\theta_P}(\mathcal{H}_{s,t-1}, \mathbf{p}^{\text{new}}_{s,t-1}, e_t), y_{s,t}\big) \tag{11}$$
where $\mathbf{p}^{\text{new}}_{s,t-1} = \text{Analyst}(\mathcal{H}_{s,t-1}; \theta_A^{\text{updated}})$.

Dataset	**ASSIST2009**			**ASSIST2012**			**Eedi**		
Metrics	ACC	ACC$_{len>15}$	F1	ACC	ACC$_{len>15}$	F1	ACC	ACC$_{len>15}$	F1
CIKT-Llama3.1-8B	**0.775**	**0.781**	**0.827**	**0.774**	**0.784**	**0.847**	**0.770**	**0.775**	**0.834**
train									
w/o Iteration & Cooperation	0.766	0.772	0.818	0.767	0.770	0.841	0.768	0.747	0.830
w/o Iteration	0.772	0.777	0.824	0.756	0.755	0.833	0.755	0.767	0.830
w/o Iteration & Distillation	0.770	0.778	0.817	0.756	0.767	0.834	0.756	0.756	0.812
w/o Distillation	0.762	0.772	0.826	0.771	0.780	0.845	0.767	0.771	0.831
inference									
w/o Profile	0.755	0.750	0.805	0.760	0.765	0.835	0.760	0.700	0.830
CIKT-Qwen2.5-7B	0.777	**0.778**	0.820	**0.780**	**0.790**	**0.852**	**0.777**	**0.781**	**0.836**
train									
w/o Iteration & Cooperation	0.765	0.770	0.812	0.766	0.777	0.843	0.761	0.756	0.826
w/o Iteration	0.768	0.772	0.814	0.775	0.747	0.851	0.775	0.771	0.833
w/o Iteration & Distillation	0.780	0.756	0.832	0.775	0.779	0.834	0.767	0.767	0.823
w/o Distillation	**0.783**	0.767	**0.832**	0.777	0.785	0.849	0.774	0.776	0.833
inference									
w/o Profile	0.749	0.750	0.798	0.768	0.732	0.844	0.765	0.757	0.835

Table 3: Results of the ablation experiments. "CIKT w/o Distillation" removes the initial GPT-4o profiling. "CIKT w/o Iteration" removes the iterative refinement but keeps the cooperative structure trained in a single pass. "CIKT w/o Iteration & Distillation" combines the above two settings (i.e., no iteration and no distillation). "CIKT w/o Iteration & Cooperation" removes both iteration and the cooperative structure, representing a more direct LLM fine-tuning. "CIKT w/o Profile (Inference)" means the full model was trained, but profiles were withheld from the Predictor during inference.

This entire cycle is repeated iteratively, fostering mutual improvements in both the Analyst and Predictor components.

4 Experiments

To systematically evaluate the efficacy, robustness, and contributions of key components within our proposed collaborative knowledge tracing framework based on large language models, this section details a series of comprehensive experiments. These experiments are designed to thoroughly investigate and address the following core research questions:

- **RQ1: Overall Performance**
  Can our CIKT surpass traditional knowledge tracing methods and other state-of-the-art large language model baselines?
- **RQ2: Ablation Study**
  What are the specific impacts of core design elements affecting overall predictive performance? What are the actual contributions of each component?
- **RQ3: Sensitivity Analysis**
  How do two critical parameters, the total number of iteration rounds and the sample size used per iteration, concretely affect the final predictive performance, and what degree of sensitivity does the model exhibit to variations in these parameters?
- **RQ4: Explainability**
  Do the user profiles generated by the Analyst demonstrably enhance the explainability of student knowledge states, and do they offer effective guidance for the Predictor's subsequent outcome predictions?

4.1 Experimental Setup

4.1.1 Datasets

Our framework was evaluated on three widely used public educational datasets and one programming dataset to assess cross-domain generalizability, providing extensive student interaction records for robust KT model training and validation:

- **ASSIST09** (Feng et al., 2009): Collected from the ASSISTments online mathematics tutoring system (2009–2010). We utilized the combined version, common in KT research.
- **ASSIST12** (Feng et al., 2009): Also from the ASSISTments platform, this dataset contains student interaction data from 2012–2013.
- **Eedi** (Wang et al., 2020): Sourced from the Eedi mathematics platform as part of the NeurIPS 2020 Education Challenge. We used the *train_task_1_2.csv* file from this challenge, adopting the leaf nodes of its provided math concept tree as the relevant knowledge concepts (KCs) for each question.
- **BePKT** (Zhu et al., 2022): Collected from an online programming Online Judge (OJ) platform, this dataset targets programming knowledge tracing and includes students' interactions with cod-

ing problems. As a domain inherently different from mathematics, it involves distinct cognitive and task demands, enabling us to evaluate the generalizability of our approach beyond math-focused KT.

Prior to model training, the datasets underwent the following preprocessing steps:

- **Data Cleaning:** Invalid or duplicate records, such as those lacking essential information (e.g., student/question IDs, response correctness), were removed.
- **User Interaction Sequence Construction:** Student records were organized into chronological interaction sequences, each detailing the assessed knowledge concepts (KCs), question difficulty, and response correctness.
- **Question Difficulty Calculation:** Question difficulty was estimated as $1 - \text{pass rate}$, derived from the average correctness for each question within the training set.
- **Sequence Segmentation and Filtering:** Complete student interaction sequences were segmented into 50-interaction subsequences; those with fewer than 5 interactions were removed to ensure effective modeling length.

Detailed statistics of the preprocessed datasets are summarized in Table 2.

4.1.2 Backbone

In our framework, both the Analyst and Predictor components leverage large language models as their backbone. We experimented with two primary models: Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct. All experiments reported in this paper were conducted on a single NVIDIA A100 GPU.

4.1.3 Evaluation

For model evaluation, each dataset was partitioned into training, validation, and test sets using an 8:1:1 ratio. As KT is a binary classification task, we employed ACC and the F1-score as standard evaluation metrics. To specifically assess framework scalability and performance on longer interaction sequences, we additionally computed ACC for subsequences where the total number of interactions (including the item to be predicted) exceeds 15, denoted as $\text{ACC}_{len>15}$. The model achieving the best validation set performance was subsequently used for test set evaluation, with prediction results assessed at the final position of each processed sequence. We take the average of five times on the test set as the displayed result. The detailed impact of sequence length characteristics, including the $\text{ACC}_{len>15}$ metric, on model performance is further discussed in the RQ1: Overall Performance section.

4.1.4 Baselines

To comprehensively evaluate our CIKT framework, we compare its performance against a diverse set of nine baselines, covering both mainstream DLKT methods and general-purpose LLMs. The selected DLKT models include pioneering approaches such as DKT (Piech et al., 2015) and the memory-augmented DKVMN (Zhang et al., 2017); Transformer-based architectures like SAKT (Pandey and Karypis, 2019) and AKT (Ghosh et al., 2020); LPKT (Shen et al., 2021), which explicitly models learning and forgetting dynamics; and other established methods IKT (Minn et al., 2022) and DIMKT (Shen et al., 2022). This group represents a spectrum of well-regarded techniques in the KT field. Furthermore, to benchmark against general LLM capabilities when applied directly to the knowledge tracing task, we include GPT-4o (Hurst et al., 2024) and Deepseek R1 (Guo et al., 2025) as LLM baselines.

4.2 RQ1: Overall Performance

To address our first research question (RQ1) concerning the predictive efficacy of our proposed framework, this section compares our CIKT against traditional KT and several LLM baselines. Detailed performance metrics across the ASSIST2009, ASSIST2012, Eedi, and BePKT are presented in Table 1.

Our CIKT framework, particularly the CIKT-Qwen2.5-7B variant, demonstrates highly competitive performance, significantly surpassing the strongest traditional KT baselines in terms of Accuracy (ACC) and F1-score across all four datasets. Notably, this performance advantage of CIKT is often more pronounced on longer interaction sequences. Such superior performance on extended histories underscores CIKT's enhanced capability to effectively model long-range dependencies and leverage comprehensive contextual information through its dynamic student profiling mechanism. Furthermore, CIKT showcases a clear advantage over general-purpose LLM baselines like GPT-4o and Deepseek-R1 when applied directly to the KT task. Their considerably lower performance highlights the necessity of CIKT's specialized, structured, iterative, and collaborative architecture, which features explicit student profile generation and targeted optimization, as opposed to direct, unspecialized LLM application.

These findings effectively address RQ1, demonstrating that our framework achieves superior performance in knowledge tracing compared to both traditional methods and direct LLM applications.

4.3 RQ2: Ablation Study

To investigate the individual contributions of key components within our CIKT framework—namely, the iterative optimization, the cooperative Analyst–Predictor structure with profile generation, and the utilization of profiles at inference—we conducted a series of ablation studies, thereby addressing RQ2. These experiments were performed using both Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct as backbone models. The specific configurations and detailed results of these studies are presented in Table 3, with the caption of the table defining each ablated variant. To address reproducibility concerns about the construction of initial profiles, we clarify that the human filtering stage was a minimal sanity check restricted to correcting objective formatting errors (e.g., typos, mismatched labels) and removing cases where the profile text contradicted the correctness label; no subjective judgments or complex annotation rubric were involved.

The results in Table 3 consistently demonstrate the critical importance of each evaluated component across both backbone models and all datasets. Comparing the full CIKT framework to **CIKT w/o Iteration**, a clear performance drop is observed, highlighting the positive impact of the iterative process on refining profile quality and enhancing Analyst–Predictor synergy. When both iteration and the cooperative profiling structure are removed (**CIKT w/o Iteration & Cooperation**), representing a direct LLM fine-tuning approach, performance generally degrades further, underscoring the benefits of CIKT's explicit two-component architecture and profile-based modeling. Most notably, the **CIKT w/o Profile (Inference)** configuration, where profiles are withheld from the Predictor during inference after full CIKT training, results in the most significant performance deterioration, confirming the crucial role of dynamically generated student profiles. Finally, the distillation-related ablations show that while external distillation helps warm-start CIKT, omitting it (**w/o Distillation**) leads to only marginal drops in ACC/F1; the Analyst–Predictor loop still self-refines useful profiles over iterations, enabling a practical trade-off between external LLM cost and the number of refinement rounds without sacrificing robustness. The results address RQ2.

4.4 RQ3: Sensitivity Analysis

To address RQ3, we analyzed model sensitivity to the total number of iteration rounds and the iteration sample size, denoted as k. These results are visually summarized in Figure 2.

First, our examination of iteration rounds, varied from 0 to 3 for the CIKT-Qwen2.5-7B model on the ASSIST2012 dataset and depicted in Figure 2(a), revealed that performance generally stabilized or reached a strong point around three iterations. This underscores the efficacy of progressive refinement in enhancing the model's predictive capabilities.

Next, regarding sensitivity to the iteration sample size k, which was varied from 500 to 2000 using CIKT-Qwen2.5-7B on the ASSIST2009 dataset as shown in Figure 2(b), results indicated that while larger k values generally improved overall metrics, these gains diminished at the higher end of the tested range. Notably, ACC on longer sequences, specifically the $\text{ACC}_{len>15}$ metric for sequences with more than 15 interactions, exhibited more sustained improvement with increasing k. This suggests a particular advantage for modeling long-term dependencies. Considering the trade-off between overall efficacy and computational overhead, an iteration sample size of $k = 1000$ was adopted for most other experiments.

4.5 RQ4: Explainability and Profile Utility

RQ4 evaluates (i) the explainability offered by our CIKT framework and (ii) the predictive utility of the user profiles generated by the Analyst. The dynamic textual profiles produced by the Analyst are intended to provide human-understandable summaries of student learning patterns, concept mastery, and areas needing attention. These profiles are further refined through the iterative process to yield progressively more nuanced, pedagogically meaningful insights that can be inspected by instructors and learning scientists.

To offer qualitative evidence, Section C presents a detailed case study demonstrating how profile refinement via iteration translates into clearer representations of students' strengths, misconceptions, and evolving mastery. The case also illustrates that such refined profiles not only enhance interpretabil-

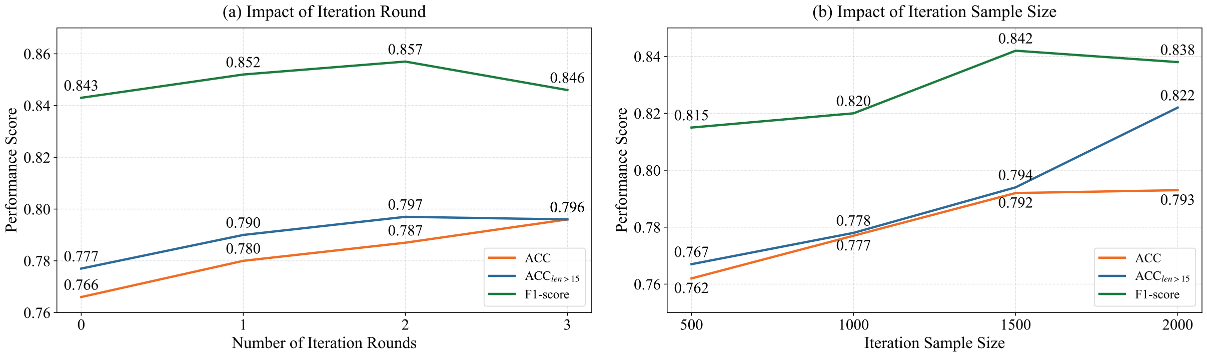

Figure 2: Effect of the number of interaction rounds and sample size per iteration on the performance of CIKT.

ity for human stakeholders, but also serve as more informative signals for the Predictor, enabling more faithful and transparent forecasting.

Beyond qualitative analysis, we conducted a blind human evaluation to quantitatively validate the utility of the generated profiles for explainability. Two expert annotators performed a forced-choice comparison on 100 profile pairs (initial vs. refined) produced by CIKT-Qwen2.5-7B on the ASSIST2009 dataset, deciding in each case which profile was better. The preferences were highly consistent: both annotators independently agreed that the refined profile was superior in 82% of the cases. Inter-annotator reliability was substantial, with Cohen's Kappa coefficient (Wan et al., 2015) $\kappa = 0.85$, providing robust quantitative evidence that the refined profiles are indeed more informative and understandable for human evaluators. The full agreement breakdown is summarized in the confusion matrix below.

A2 \ A1	Refined Better	Initial Better
Refined Better	82	1
Initial Better	3	14

Table 4: Agreement matrix for profile quality (A1=Annotator 1, A2=Annotator 2).

In terms of predictive utility, the refined profiles provide the Predictor with more targeted and stable signals about students' evolving mastery, which leads to enhanced forecast accuracy and improved interpretability of the model's decision pathways. Taken together, the qualitative case evidence and the quantitative blind evaluation jointly demonstrate that CIKT delivers explainable, human-usable profiles that substantively contribute to more transparent and effective prediction, thereby addressing both aspects of RQ4.

5 Conclusion

We proposed CIKT to iteratively optimize student profiling and performance prediction for accurate and explainable knowledge tracing. Its synergistic architecture enables continuous mutual enhancement, yielding significantly improved predictive accuracy and explainable student profiles on multiple educational datasets.

Limitations

Because our framework primarily generates binarized judgments for student responses, we focused on metrics such as ACC and F1-score, and consequently did not employ ranking-sensitive evaluation metrics like AUC. Moreover, due to the context window constraints of our backbone models, we did not incorporate the textual content of question stems, which limited the potential for more fine-grained, content-aware modeling of student knowledge.

Acknowledgments

This work was supported in part by National Key R&D Program of China (No. 2023YFC3341200) and National Natural Science Foundation of China (No. 92270119).

References

Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. 2023. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14196–14204.

Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278.

Jiajun Cui, Minghe Yu, Bo Jiang, Aimin Zhou, Jianyong Wang, and Wei Zhang. 2024. Interpretable knowledge tracing via response influence-based counterfactual reasoning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1103–1116. IEEE.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19:243–266.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Yoonjin Im, Eunseong Choi, Heejin Kook, and Jongwuk Lee. 2023. Forgetting-aware linear bias for attentive knowledge tracing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3958–3962.

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

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. 2023. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85.

Sein Minn, Jill-Jênn Vie, Koh Takeuchi, Hisashi Kashima, and Feida Zhu. 2022. Interpretable knowledge tracing: Simple and efficient student modeling with causal relations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 12810–12818.

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*, pages 156–163.

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. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1205–1214.

Zachary A Pardos and Neil T Heffernan. 2010. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings 18*, pages 255–266. Springer.

Zachary A Pardos and Neil T Heffernan. 2011. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization: 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings 19*, pages 243–254. Springer.

Philip I Pavlik, Hao Cen, and Kenneth R Koedinger. 2009. Performance factors analysis–a new alternative to knowledge tracing. In *Artificial intelligence in education*, pages 531–538. Ios Press.

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.

Edward O Pyzer-Knapp, Jed W Pitera, Peter WJ Staar, Seiji Takeda, Teodoro Laino, Daniel P Sanders, James Sexton, John R Smith, and Alessandro Curioni. 2022. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. *npj Computational Materials*, 8(1):84.

Richard Scruggs, Ryan S Baker, and Bruce M McLaren. 2019. Extending deep knowledge tracing: Inferring interpretable knowledge and predicting post-system performance. *arXiv preprint arXiv:1910.12597*.

Shuanghong Shen, Zhenya Huang, Qi Liu, Yu Su, Shijin Wang, and Enhong Chen. 2022. Assessing student's dynamic knowledge state by exploring the question difficulty effect. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 427–437.

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.

Keith Tyser, Ben Segev, Gaston Longhitano, Xin-Yu Zhang, Zachary Meeks, Jason Lee, Uday Garg, Nicholas Belsten, Avi Shporer, Madeleine Udell, et al. 2024. Ai-driven review systems: evaluating llms in scalable and bias-aware academic reviews. *arXiv preprint arXiv:2408.10365*.

TANG Wan, HU Jun, WU Pan, HE Hua, et al. 2015. Kappa coefficient: a popular measure of rater agreement. *Shanghai archives of psychiatry*, 27(1):62.

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.

Jing Wang, Huifang Ma, Mengyuan Zhang, Lei Zhang, and Liang Chang. 2025. Multi-granularity ensemble interaction graph modeling for knowledge tracing. *Knowledge-Based Systems*, 309:112834.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024a. Scimon: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. 2024b. Autosurvey: Large language models can automatically write surveys. *Advances in Neural Information Processing Systems*, 37:115119–115145.

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, et al. 2020. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*.

Huali Yang, Junjie Hu, Jinjin Chen, Shengze Hu, Jing Geng, Qiang Zhu, and Tao Huang. 2025. Mahkt: Knowledge tracing with multi-association heterogeneous graph embedding based on knowledge transfer. *Knowledge-Based Systems*, 310:112958.

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*, pages 765–774.

Jinjin Zhao, Shreyansh Bhatt, Candace Thille, Dawn Zimmaro, and Neelesh Gattani. 2020. Interpretable personalized knowledge tracing and next learning activity recommendation. In *Proceedings of the seventh ACM conference on learning@ scale*, pages 325–328.

Renyu Zhu, Dongxiang Zhang, Chengcheng Han, Ming Gaol, Xuesong Lu, Weining Qian, and Aoying Zhou. 2022. Programming knowledge tracing: A comprehensive dataset and a new model. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 298–307. IEEE.

A Notation Table

We list and explain the notations in our methodology in Table 5.

B Hyper-parameter Setting

We provide the training and inference hyper-parameter settings in Table 6 and Table 7.

C Case Study

To qualitatively illustrate the efficacy of our CIKT framework's iterative optimization process, particularly its impact on the quality of student profiles generated by the Analyst and the subsequent prediction accuracy of the Predictor, we present a detailed case study. The selected case, detailed in Table 8, involves a student interaction sequence where the Predictor's outcome for the "Next Question" was incorrect based on the initial profile generated by the Analyst before iteration, but became correct after the Analyst was refined through iterations. The ground truth for the "Next Question" – (['Conversion of Fraction Decimals Percents'], 0.16) – was "False".

Profile Before Iteration and Initial Prediction. The initial user profile generated by the Analyst before iterative refinement is presented in the left column of the bottom table in Table 8. This profile, while attempting to summarize performance across concepts like "Making a Table from an Equation," "Equivalent Fractions," and "Conversion of Fraction Decimals Percents," tends to exhibit characteristics of a more direct translation or surface-level summary of the interaction sequence. For instance, it meticulously lists the number of attempts and correctness for each topic (e.g., "The student has encountered three questions related to this topic, all answered incorrectly" for "Making a Table from

Notation	**Description**
General Notations	
s	Index for a student
e_i	The i-th exercise or question encountered by a student
r_i	Binary correctness (0 or 1) of the student's response to e_i
$\mathcal{S}_s$	Historical interaction sequence for student s, e.g., $\mathcal{S}_s = \{(e_1, r_1), \ldots, (e_N, r_N)\}$
$\mathbf{p}_s$	User profile (typically textual) for student s, generated by the Analyst
θ_A	Parameters of the Analyst model
θ_P	Parameters of the Predictor model
$\mathcal{D}_{\text{train}}$	Set of training instances (e.g., students or student-interaction sequences)
$\mathcal{L}_{\text{CE}}(\cdot, \cdot)$	Standard cross-entropy loss function
Stage 1: Distillation	
$\text{LLM}_{\text{teacher}}$	Large-parameter teacher model (e.g., GPT-4o) for initial profile annotation
$\mathbf{p}_{s,\text{teacher}}$	Initial textual profile for student s generated by $\text{LLM}_{\text{teacher}}$ (Eq. 1)
$\mathbf{p}^*_{s,\text{teacher}}$	Curated, high-quality teacher profile for student s used for training
$\mathbf{p}_{s,\text{analyst}}$	Profile for student s generated by the Analyst during distillation (Eq. 2)
$\mathcal{L}_{\text{Distill}}(\theta_A)$	Distillation loss for training the Analyst (Eq. 3)
Stage 2: Profiling	
$\{\mathbf{p}_s\}$	Set of all generated user profiles
$\{\mathcal{S}_s\}$	Set of all student interaction sequences
Stage 3: Reasoning	
$\mathcal{H}_{s,t-1}$	Historical interaction sequence for student s up to interaction $t-1$
$\mathbf{p}_{s,t-1}$	User profile for student s based on history $\mathcal{H}_{s,t-1}$
e_t	The t-th (current or next) exercise for student s
$y_{s,t}$	Actual binary correctness of student s's response to e_t
$\hat{y}_{s,t}$	Predicted probability of student s correctly answering e_t (Eq. 5)
$\mathcal{L}_{\text{Predict}}(\theta_P)$	Predictive loss for training the Predictor (Eq. 6)
Stage 4: Iteration	
$\mathbf{x}_t$	A student's historical interaction sequence
π_{θ_A}	The Analyst viewed as a policy parameterized by θ_A
$\mathbf{p}_t$	User profile generated by Analyst π_{θ_A} from $\mathbf{x}_t$ (Eq. 7)
f_{θ_P}	The Predictor model parameterized by θ_P
e_{t+1}	The subsequent exercise for which a prediction is made based on $\mathbf{x}_t, \mathbf{p}_t$
y_{t+1}	Ground truth outcome for the prediction $\hat{y}_{t+1}$
$\hat{y}_{t+1}$	Prediction by f_{θ_P} for e_{t+1}
r_{t+1}	Binary reward signal ($+1$ or -1) based on prediction accuracy (Eq. 9)
$\mathcal{L}_{\text{KTO}}(\theta_A)$	KTO loss function for optimizing the Analyst (Eq. 10)
$\log \pi_{\theta_A}(\mathbf{p}_t \mid \mathbf{x}_t)$	Log-probability of the Analyst generating profile $\mathbf{p}_t$ given sequence $\mathbf{x}_t$
$\theta_A^{\text{updated}}$	Updated parameters of the Analyst after a KTO step
$\mathbf{p}^{\text{new}}_{s,t-1}$	Profile generated by the updated Analyst using history $\mathcal{H}_{s,t-1}$

Table 5: Notation Table for the CIKT Framework.

Parameter(training)	Analyst	Predictor
# lora_rank	8	8
# learning_rate	5.0e-6	1.0e-4
# train_epochs	10	10
# warmup_ratio	0.1	0.1

Table 6: Training hyper-parameter setting of CIKT.

Parameter(inference)	Analyst	Predictor
# temperature	0.95	0
# top_p	0.7	0.7
# top_k	50	50

Table 7: Inference hyper-parameter setting of CIKT.

an Equation"). While it provides some overview, it contains redundancies in its factual recounting and offers a somewhat limited depth of analytical insight beyond stating observed patterns.

Crucially, regarding the "Next Question" on "Conversion of Fraction Decimals Percents" (difficulty 0.16), the profile (as highlighted in Red notes: "- *This question is slightly easier than the previous one, which may provide an opportunity for the student to consolidate their understanding.*" This particular phrasing, emphasizing the "easier" nature and "opportunity to consolidate," might lead the Predictor to infer a higher likelihood of a correct answer. In this instance, the Predictor, relying on this pre-iteration profile, made an incorrect prediction (implicitly predicting "True", while the real response was "False").

Profile After Iteration and Corrected Prediction. The right column of the bottom table in Table 8 showcases the student profile generated by the Analyst after several KTO iterations. This refined profile demonstrates a notable improvement in several aspects. It is more focused in its analysis, moving beyond simple sequence translation to offer more structured insights and actionable recommendations. For example, it categorizes observations into "Overall Performance and Patterns," "Difficulty and Learning Progression," "Projected Next Question," and detailed "Recommendations." This structure itself provides a clearer, more pedagogically useful summary. The suggestions provided, such as "Focus on reinforcing understanding of "Table" as a standalone topic..." and "Work on integrating concepts...", are more specific and reliable, offering genuine guidance applicable in real educational scenarios.

The shift in the analysis of the "Next Question" is particularly significant. The refined profile highlighted in Green states: "- *The upcoming question on "Conversion of Fraction Decimals Percents" with a difficulty of 0.16 is consistent with the difficulty level at which the student has previously struggled with this knowledge point, presenting a challenge.*" This revised perspective, informed by the iterative feedback loop, correctly identifies the question as a challenge despite its slightly lower difficulty, considering the student's prior struggles with the same concept at a similar difficulty (0.17, as per the profile's detailed breakdown). This more nuanced and context-aware assessment likely guided the Predictor to correctly forecast the outcome as "False" for the "Next Question", aligning with the ground truth.

Question and Input
Question: The student has completed some questions in a listing order from earliest to latest, with each ternary in the subsequent sequence representing a piece of information containing the knowledge points contained in the question, the difficulty of the question (expressed as the complement of the pass rate on the full set), and the response to the question, whether true or false. The sequence culminates with the presentation of a binary comprising the knowledge points and the difficulty of the subsequent question. Please provide a high-level summary of the student's state of knowledge acquisition in detail. **Input:** The student's historical response sequence: (['Making a Table from an Equation'], 0.47, False), (['Making a Table from an Equation'], 0.64, False), (['Making a Table from an Equation'], 0.61, False), (['Equivalent Fractions'], 0.13, True), (['Equivalent Fractions'], 0.26, True), (['Equivalent Fractions'], 0.19, True), (['Equivalent Fractions'], 0.37, False), (['Equivalent Fractions'], 0.32, False), (['Equivalent Fractions'], 0.39, False), (['Conversion of Fraction Decimals Percents'], 0.17, True), (['Conversion of Fraction Decimals Percents'], 0.17, False) **Next Question:** (['Conversion of Fraction Decimals Percents'], 0.16) **Real response(No input):** False

Profile Before Iteration	**Profile After Iteration**
1. **Making a Table from an Equation**: - The student has encountered three questions related to this topic, all answered incorrectly. - The difficulty levels of these questions ranged from 0.47 to 0.64, indicating a moderate level of challenge. - This suggests that the student may not yet have a solid understanding of this concept or is struggling with the application of this knowledge. 2. **Equivalent Fractions**: - The student has answered five questions related to this topic, with a mix of correct and incorrect responses. - The difficulty levels ranged from 0.13 to 0.39. - The student demonstrated a good grasp of easier questions (e.g., those with a lower difficulty, such as 0.13 and 0.26), but struggled with more challenging ones (e.g., 0.37, 0.32, and 0.39). - This indicates that while the student has some understanding of the topic, there may be gaps in knowledge or difficulty with more complex problems. 3. **Conversion of Fraction Decimals Percents**: - The student has attempted two questions related to this topic. - The first question was answered correctly, and the second was answered incorrectly. - Both questions had a difficulty level of 0.17. - The student shows potential understanding of this topic but may need further practice or reinforcement to solidify their knowledge. **Next Question**: - The next question is related to "Conversion of Fraction Decimals Percents" with a difficulty level of 0.16. - This question is slightly easier than the previous one, which may provide an opportunity for the student to consolidate their understanding. **Overall Summary**: - The student demonstrates varying levels of understanding across different topics. - They show strong potential in "Equivalent Fractions" for easier questions but may need more practice with more challenging problems. - They seem to have a good understanding of "Conversion of Fraction Decimals Percents" but need reinforcement. - There is a clear area for improvement in "Making a Table from an Equation," as the student has consistently answered questions incorrectly, indicating a need for focused practice and possibly revisiting foundational concepts.	1. **Making a Table from an Equation:** - The student initially struggled with questions related to "Making a Table from an Equation," as evidenced by incorrect responses to questions with moderate difficulty levels (0.47, 0.64, and 0.61). This suggests a gap in understanding or application of this concept. 2. **Equivalent Fractions:** - The student demonstrated initial proficiency with "Equivalent Fractions," correctly answering questions with low to moderate difficulty levels (0.13 and 0.26). However, there was a decline in performance as the difficulty increased, with incorrect responses to questions at 0.37, 0.32, and 0.39. This indicates that while the student has a foundational understanding, they may struggle with more complex applications or problem variations of this concept. 3. **Conversion of Fraction Decimals Percents:** - The student showed mixed results with "Conversion of Fraction Decimals Percents." They correctly answered a question with a low difficulty level (0.17), indicating a basic understanding, but then responded incorrectly to another question of the same difficulty level (0.17). This inconsistency suggests that the student may need further practice or clarification to solidify their understanding of this concept. 4. **Overall Knowledge Acquisition:** - The student's knowledge acquisition shows a pattern of initial difficulty with a concept, followed by some improvement or partial understanding, but often ending with inconsistencies or gaps in understanding, especially as the difficulty increases. - The upcoming question on "Conversion of Fraction Decimals Percents" with a difficulty of 0.16 is consistent with the difficulty level at which the student has previously struggled with this knowledge point, presenting a challenge. 5. **Recommendations:** - The student should focus on reinforcing their understanding of "Making a Table from an Equation" and "Equivalent Fractions" through targeted practice and review of fundamental concepts. - For "Conversion of Fraction Decimals Percents," the student should review the basic principles and practice a variety of problems to build confidence and consistency in their responses. - Providing feedback on incorrect responses and ensuring the student understands the reasoning behind the correct answers will be crucial in improving their overall performance. - Gradually increase difficulty level in practiced areas.

Table 8: Example of profiles before and after iteration.