ReAL: How Can LLMs Simulate the Real Teacher? Retrieval-enhanced Agent for Adaptive Learning

Rui Lv[1], Qi Liu[1,2*], Weibo Gao[1], Jiatong Li[1], Kai Zhang [1], Shiwei Tong [3]

[1]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]Tecent Company
lvrui2018@mail.ustc.edu.cn, qiliuql@ustc.edu.cn, weibogao@mail.ustc.edu.cn,
cslijt@mail.ustc.edu.cn, kkzhang08@ustc.edu.cn, shiweitong@tencent.com

Abstract

Adaptive learning focuses on recommending personalized materials (e.g., exercises, courses) to the unique needs of learners. Despite significant research, these methods still lag behind real teachers including two main limitations: (1) Prior methods model learner-item interactions based only on ID sequences, leading to insufficient use of both learner and item information, particularly the inability to leverage semantic content from item text; (2) The data-driven reinforcement learning frameworks struggle with stable performance in scenarios with sparse learning logs. To address these challenges, we introduce the Retrieval-enhanced Agent for Adaptive Learning (ReAL) powered by large language models (LLMs), to simulate teacher decision-making with extensive prior knowledge and teaching experience. Specifically, we approach the simulation from both internal and external perspectives. From the internal perspective, we utilize the superior natural language standing ability of LLMs to analyze item texts and learner profiles. This mechanism contributes to the generation of personalized and appropriate item candidates. From the external perspective, we simulate the teacher experience by retrieving similar learners, further ensuring the model's performance on sparse interaction data. Furthermore, we design a reflector based on learners' feedback to refine the recommendation process. Evaluation on three real-world datasets demonstrates the superiority of ReAL in both data utilization, recommendation accuracy and stability compared to various representative baselines.

1 Introduction

Unlike traditional classroom teaching, which provides the same materials for all learners, adaptive learning offers personalized tasks (e.g., exercises) and pathways. This makes adaptive learning a more efficient approach (Corbett, 2001; Liu et al., 2019; Huang et al., 2019). Since individual human tutoring is expensive, computer-based adaptive learning methods, also known as Learning Path Recommendation (LPR) methods, have been widely studied. Earlier works employed traditional recommendation algorithms or deep learning-based methods to suggest similar learning paths for learners (Elshani and Nuçi, 2021; Nabizadeh et al., 2020a). But these approaches often struggle to handle complex and dynamic learning processes. The state-of-the-art approaches currently model the LPR task as a Markov decision process and use reinforcement learning to train the recommendation policy, in order to improve the learner's state in the target knowledge concept (Liu et al., 2019; Li et al., 2023b, 2024a,b). However, these methods still lag behind real teachers and have two main limitations: (1) **Insufficient information utilization**: Current methods merely use the IDs to model the learner-item interaction and do not leverage the rich semantic content within the text of learning items, which results in limited performance. The text content implicitly reveals various attributes, such as the knowledge concepts covered and cognitive de-

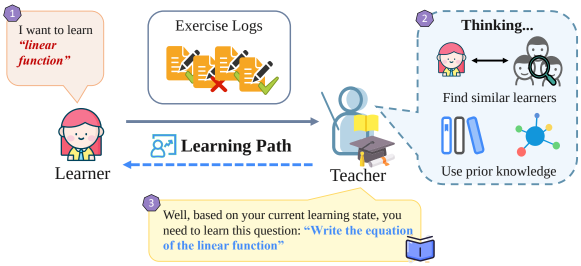

Figure 1: The process of adaptive learning with an experienced teacher involves the following steps: (1) The learner has a learning target, such as "linear functions", which represents a knowledge concept. (2) The teacher devises a personalized learning strategy based on their prior knowledge and teaching experience with similar learners. (3) Teacher recommends the next exercise to help the learner master the targeted knowledge concept.

[*]Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 168–181

mands, which are difficult to capture effectively with existing methods. For example, while both exercises assess the concept of *"linear functions"*, exercise *A: "Write the equation of the linear function"* focuses only on basic concepts. In contrast, exercise *B: "Given points A(-1, 0) and B(0, -3), write the equation of the line after shifting line AB downward by 2 units."* not only tests *"Linear Functions"* but also assesses deeper knowledge of *"Function translation."* While some existing methods employ knowledge graphs (Liu et al., 2019; Li et al., 2023b; Wu et al., 2024) to assist in learning path recommendations, they are constrained by the size and granularity of the knowledge graphs, leading to less accurate recommendations. (2) **Instable Performance**: These studies based on RL methods rely on abundant interaction data for effective training (Liu et al., 2019; Li et al., 2023b; Kubotani et al., 2021). However, the interaction data is often sparse in real-world online education, leading to unstable performance, even failing to improve the learner's learning state.

However, an experienced human teacher can easily address these issues, as shown in Figure 1. Teachers are able to obtain a more fine-grained understanding of the questions by analyzing the textual information. Additionally, they can leverage their extensive teaching experience to recognize the learner's profile from just a few exercise logs and recommend suitable learning paths based on similar cases from previous students. Recently, Large Language Models (LLMs), pre-trained on vast amounts of text data, have demonstrated rich prior knowledge and excellent performance on various reasoning tasks (Shi et al., 2024; Ouyang et al., 2022; Touvron et al., 2023), enabling them to infer from limited data samples. Some studies have used LLMs to construct powerful agents to simulate human experts (Han et al., 2024; Yang et al., 2024). In addition, recent advancements in Retrieval-Augmented Generation (RAG) (Gan et al., 2025; Yu et al., 2025)get successful by integrating LLMs with external information retrieval, enabling accurate generation. Inspired by this, we propose a new **R**etrieval-**e**nhanced **A**gent for **A**daptive **L**earning (ReAL) with LLMs, which collaboratively plans learning paths for learners from both **Internal** and **External** perspectives to simulate the experienced human teacher. From the internal perspective, we use the powerful language understanding of LLMs to summarize the current learner's profile from limited learning logs and recommend a set of candidate learning items. Specifically, we designed a **Planner** module to handle this, incorporating a memory component to store relevant information, which forms the basis for the recommendation strategy. To mitigate LLM hallucinations due to limited domain-specific knowledge, we supplement them with educational tools (e.g., knowledge concept graphs). From the external perspective, we simulate past students' teaching experience through the **Actor** module. Specifically, we construct multiple simulators to simulate the previous learners using static data from the training dataset. Then, we design a retrieval mechanism that provides select suggestions by retrieving learners with similar learning states. To adaptively refine the recommendation process, we design the **Reflector** module to update the recommendation strategy based on the learner's feedback. These three modules collaborate to dynamically recommend learning paths as interactions progress, creating an effective and stable adaptive learning agent. Overall, our contributions are as follows:

- To the best of our knowledge, we are the first to integrate LLMs' knowledge and previous learners' states into learning path recommendations, contributing to the exploration of LLMs' potential in educational tasks.
- We developed a novel agent system including Internal and External perspectives to simulate human-like decision-making and ensure stable performance in sparse data scenarios, which is hard to solve in existing methods.
- Extensive experiments on three real-world datasets demonstrate our superiority.

2 Related Work

2.1 Adaptive Learning

In online education, Learning Path Recommendation (LPR) is crucial for designing structured paths to help students systematically acquire knowledge and skills (Liu et al., 2019; Li et al., 2023b). Researchers have proposed various LPR methods, generally categorized into two types (Chen et al., 2023; Nabizadeh et al., 2017): (1) Complete generation, where a full path of predetermined length is generated and presented all at once, and (2) Step-by-step generation, where a dynamic path is generated in real time, adjusting based on learner feedback and previous steps (Liu et al., 2019; Li et al., 2023b; Zhang et al., 2024; Li et al., 2024b). The main drawback of complete generation is its failure to account for learners' cognitive changes during the process, leading to inefficient or unsuitable paths (Nabizadeh et al., 2020b). Step-by-step methods, which consider dynamic interactions, are gaining traction. Earlier works used traditional recommendation or deep learning methods to suggest similar paths for comparable learners (Elshani and Nuçi, 2021; Nabizadeh et al., 2020a), but these approaches rely on static sequences and struggle with complex, dynamic learning processes. More recently, LPR has been modeled as a Markov Decision Process, utilizing reinforcement learning to train recommendation strategies. While these methods show promising results, they still lag behind human teachers. They mainly model learner and item IDs and require extensive interaction data (Zhang et al., 2024; Li et al., 2024b), but real-world educational settings often face data sparsity, limiting personalization and stability. Our proposed ReAL addresses these challenges by leveraging LLMs to analyze and summarize textual information and by using simulators that recommend items based on feedback from similar learners, providing a more human-like approach.

2.2 LLM-Driven Agents in Education

LLM-empowered generative agents demonstrate impressive abilities in perceiving environments, decision-making, and action-taking, attracting significant research interest (Wang et al., 2024; Li et al., 2023a; Han et al., 2024; Yang et al., 2024). In education, these agents have opened new possibilities. For example, (Wu et al., 2023) use chat-optimized LLMs as agents in multi-agent dialogues to collaboratively solve complex queries, showing the potential to address a wide range of general questions. In learner simulation, such as Agent4Edu (Gao et al., 2025) and EduAgent (Xu et al., 2024), employ LLM-based agents to simulate learners interacting with exercises, presentations and videos, evaluating their performance by predicting quiz results. In educational recommendation, (Li et al., 2024a) leverages LLMs' factual knowledge to create SKarREC, a concept recommendation model that improves suggestions for the next concept a learner should study. It is important to note that while SKarREC focuses on recommending knowledge concepts, our task is to recommend learning exercises in a path.

3 Problem Definition

Following (Liu et al., 2019; Li et al., 2023b), We focus on step-by-step recommendations for session-based learning paths based on real-time interactions. A learner's process typically involves two types of items: learning items (e.g., knowledge concepts or skills) and exercise items (e.g., questions). Without loss of generality, we denote the knowledge concept item set as $\mathcal{KI} = \{k_1, k_2, \ldots, k_m\}$ and the exercises item set as $\mathcal{E} = \{e_1, e_2, \ldots\}$. Instead of only using item index in previous work, the $k_i \in \mathcal{KI}$ and $e_i \in \mathcal{E}$ consist of item index and item text content. The learner's goals are denoted as $\mathcal{G} = \{g_1, g_2, \ldots, g_m\}$, where $g_i \in \mathcal{KI}$. The learning process is as follows: before starting, educational tools test a learner on their goals to obtain an initial score E_s. The learning path is generated step-by-step as $\mathcal{P} = \{e_1, e_2, \ldots, e_p\}$, where $e_i \in \mathcal{E}$, aligning with previous works. After completing the entire learning path, a final test using educational tools (e.g., knowledge tracing models) is taken on the learning goals to obtain a final score E_e, allowing to calculate learning effectiveness E_p:

$$E_p = \frac{E_e - E_s}{E_{sup} - E_s}, \tag{1}$$

where E_{sup} is the total exam score, equal to the number of learning goals. Our aim is to maximize E_p by providing an effective learning path.

4 ReAL Framework

Our ReAL, shown in Figure 2, consists of three modules: Planner, Actor, and Reflector. The Planner analyzes the current learner from an internal perspective. It leverages LLMs' semantic understanding and reasoning abilities, along with educational tools, to recommend a set of candidate exercises. The Actor simulates an experienced teacher from an external perspective. It selects the most suitable item for the learner by considering the performance of similar past learners on the candidate items. Finally, the Reflector updates the recommendation strategy in the Planner based on feedback.

4.1 Planner (Internal)

The Planner module is designed to simulate a teacher with educational prior knowledge. This module consists of three parts: memory, educational tools, and LLMs. The LLM-based planner can flexibly use educational tools to enhance

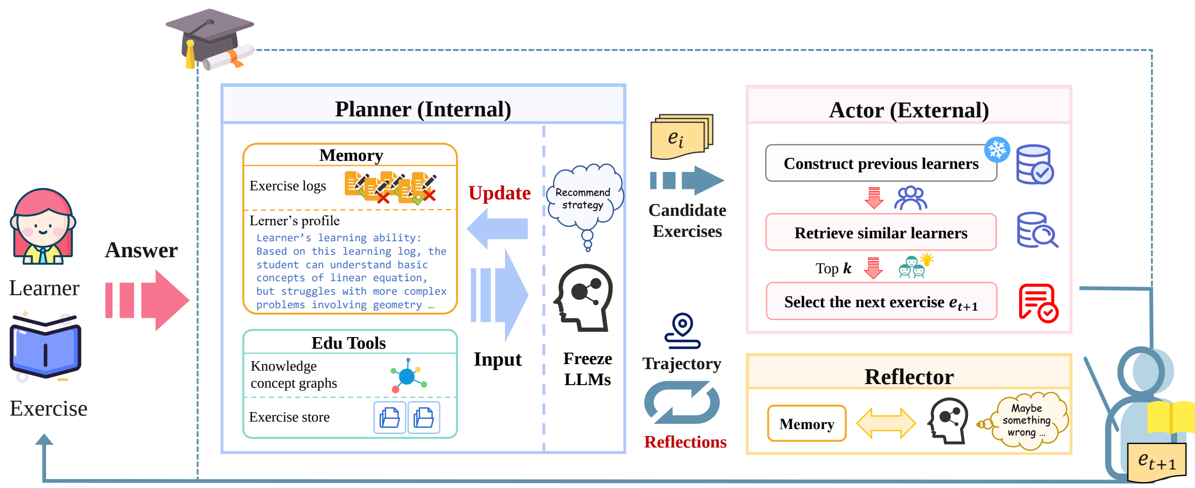

Figure 2: The overview of our ReAL framework. (1) **Planner**: Analyzes the current learner from an **Internal** perspective and generates a set of candidate exercises. (2) **Actor**: Simulates teacher experience from an **External** perspective and provides dynamic recommend suggestions (e_{t+1}) based on similar learners. (3) **Reflector**: Generate the reflections based on the learner's feedback, which is used to update the recommendation strategy.

their expertise by incorporating external educational knowledge (e.g., knowledge concept graphs). It then uses its language understanding and reasoning capabilities to summarize the learner's profile, which is stored and updated in memory. Finally, it synthesizes all information to filter a candidate set of items from the existing exercise pool.

Educational Tools

The Planner aims to retrieve the top-n suitable question candidates $\mathcal{C} = \{e_i, ..., e_{i+n}\}$ from the question bank, tailored to the learner's target knowledge concept. It leverages all available information from memory and the recommendation strategy provided by the Reflector. We need to employ educational prior knowledge to narrow the search space. Furthermore, While LLMs have strong general knowledge, they often exhibit limitations in specific domains and can generate hallucinations (Zhang et al., 2023). Therefore, external knowledge is necessary to guide and assist LLMs in decision-making (Lewis et al., 2020; Chen et al., 2024). Based on this, we design educational tools to provide domain-specific knowledge. Specifically, we use a hierarchical knowledge graph $\mathcal{G}$ to restrict retrieval to the current knowledge concept and its immediate predecessors. The exercises linked to this concept form the exercise set $\mathcal{S}_t \subseteq \mathcal{E}$.

Memory

The memory is designed to store the learning history and the feedback from the LLMs, i.e., the current learner's profile. Specifically, all item-feedback pairs construct the historical learning record, denoted as $\mathcal{H} = (e, answer)$. After time t, $(e_t, answer_t)$ is added to the historical record, i.e., $\mathcal{H}_t = \mathcal{H}_{t-1} \cup (e_t, answer_t)$. The data in time(step) t of the memory $\mathcal{M}_t^I$ can be presented as follow:

$$\mathcal{M}_t^I \leftarrow \mathcal{H}_t, \tag{2}$$

Planning

In this process, we perform two steps: (1) summarize the learner's profile; (2) provide a candidate set of recommended items using educational tools.

- We use LLMs to generate the learner's learning profile based on the learner's response records, including learning ability and learning preferences. At time t, the learner's profile L_t can be note as:

$$L_t \sim \text{LLM}(\mathbf{P_L}, \mathcal{H}_t), \tag{3}$$

where $\mathcal{H}_t$ is the historical learning logs in $\mathcal{M}_t^I$, and $\mathbf{P_L}$ is the prompt to generate the summaries and designed as follows,

> *You're a seasoned math teacher with ten years of teaching experience. Please use one sentence to summarize the student's learning ability and learning preference from the following learning logs:[...].*

The memory $\mathcal{M}_t^I$ can be updated by:

$$\mathcal{M}_t^I \leftarrow \mathcal{M}_t^I \cup L_t. \tag{4}$$

- Now, LLM can predict the candidate exercises $\mathcal{C}$:

$$\mathcal{C} \sim \text{LLM}(\mathbf{P_C}, \mathcal{M}_t^I, \mathcal{S}_t, R_t), \tag{5}$$

where $\mathbf{P_C}$ is the LLM prompt, and R_t represents reflections from the Reflector, which include the recommendation strategy at step t. The prompts we designed are as follows,

> *Given the following history text [...] and the recommend reflection [...], the student profile: [...], the knowledge learning goal [...]. Here is the candidate exercise list: [...]. Please provide the k suitable exercises from the above list to help the student achieve the learning goal efficiently.*

4.2 Actor (External)

In the Actor module, we simulate teaching experiences from past students from an external perspective. We first create multiple simulators to model previous learners using static data from the training dataset. Then, we design a retrieval mechanism to offer targeted suggestions by identifying learners with similar learning states.

Construct previous learners

In this module, we model learners from the training dataset as past students, simulating real teaching experiences by estimating their knowledge states. Since a learner's exact knowledge state is not directly observable, prior work developed cognitive diagnosis (CD) (Gao et al., 2021) and knowledge tracing (KT) (Shen et al., 2024) methods. CD assesses the learner's mastery of specific knowledge concepts, providing detailed feedback on areas of weakness. KT tracks the learner's learning progress over time, predicting future performance based on past learning activities. These educational tools provide valuable references for constructing learners' states. In our framework, we use a pre-trained deep knowledge tracing model (DKT) to output learners' mastery states. It's important to note that these educational tools are just alternatives, and developing them isn't our main contribution. The datastore is constructed as follows:

$$\mathbf{D} = \{(\mathbf{k}_i, \mathbf{v}_i) | 1 \leq i \leq N\}, \tag{6}$$

where $\mathbf{k}_i = \{s_1, ..., s_m\}$, $s_j \in [0, 1]$, is generated by DKT model, representing the ith learner's mastery states of every knowledge concept. $\mathbf{v}_i$ is the corresponding learning logs, including the items and response. N is the number of learners in the training dataset.

Retrieval and Selection

For the current learner, we use the same educational tool that was used to construct the datastore to obtain their knowledge state $\mathbf{k}_t$, then retrieve the top-k most similar learners from the database $\mathbf{D}$. By considering the feedback of similar learners on the candidate exercises $\mathcal{C}$, we select the most suitable problem for the current learner. The process are as follows:

- Retrieve the top-k learners from $\mathbf{D}$:

  $$\mathcal{Y}, \mathbf{W} = \mathrm{Sim}(\mathbf{k}_t, \mathbf{k_D}), \tag{7}$$

  where $\mathcal{Y} = \{(\mathbf{k}_i, \mathbf{v}_i) | 1 \leq i \leq k\}$ denotes the similar learners' key-value pairs. $\mathbf{W} \in \mathbf{R}^{1\times k}$ represents the distance between the current learner's state and these similar previous learners. Here, we use Cosine similarity to calculate Sim.

- Next, we use these similar learners to vote on the candidate exercises. Specifically, the vote value is calculated by the educational tool, which gives the learning state improvement after completing these exercises:

  $$\mathbf{V} = \mathrm{EduTool}(\mathcal{Y}, \mathcal{C}), \tag{8}$$

  where $\mathbf{V} \in \mathbf{R}^{k\times n}$ represents the learning gain of k learners across n exercises.

- Finally, we aggregate the information and select the exercise with the highest votes from the historical learners as e_{t+1}:

  $$e_{t+1} = \mathcal{C}[\arg\max_i (\mathbf{W} \times \mathbf{V})_i]. \tag{9}$$

4.3 Reflector

Reflect on the efficiency and rationality of past recommendations by comparing the learner's actual responses with the predicted responses. This step is similar to the gradient backpropagation process in neural network training, which is used to update the model parameters. At time t, the reflection R_t generation process can be formulated as:

$$R_t \sim \mathrm{LLM}(\mathbf{P_R}, \mathcal{M}_t^I), \tag{10}$$

where $\mathbf{P_R}$ notes the prompt for LLM to generate the reflection and designed as follows,

> *Please use one sentence to reflect the strategy of recommend questions from the learner's feedback:[...].*

Then, R_t is passed to the Planner to update the recommendation strategy.

Statistics	Junyi	ASSIST09	TextLog
Knowledge Concepts	36	97	698
Exercises	711	16,836	8021
Learners	245,511	4,092	127,610
Response records	25,367,573	397,235	1,680,886
Records / Learners	1034	97	13

Table 1: Statistics of datasets.

5 Experiment

5.1 Dataset

Our experiments are conducted on three real-world public datasets: Junyi[1], ASSIST09[2], and a dataset we collected, named TextLog, which includes question text content details from real-world scenarios. All datasets contain learners' learning log data and the knowledge concept names for all exercises. For the Junyi dataset, we use the "topic" field as the learning items, which are commonly used in education, and the "name" field (exercise name) as the text content of the exercises. In the ASSIST09 dataset, the "skill name" field represents the learning items. Since ASSIST09 does not provide direct information about the exercises, we use other available fields, such as "response time" and "original (Main/Scaffolding Problem)", to represent the exercises. Additionally, the TextLog dataset includes the text content for each practice item, with an example shown in the Appendix. We also construct the knowledge transition graph as described in (Gao et al., 2021) for all three datasets. The dataset statistics are provided in Table 1.

5.2 Simulators

A critical challenge in evaluation is that existing realistic datasets only provide static information, making it difficult to assess whether practice items not presented in a sequence can be answered correctly (Huang et al., 2019). As a result, these datasets are not suitable for evaluating learning paths. To address this, we follow prior works (Liu et al., 2019; Chen et al., 2023; Li et al., 2024b; Zhang et al., 2024) and employ a Knowledge Evolution-based Simulator (KES) as introduced in (Liu et al., 2019). KES is a data-driven system that utilizes the DKT model (Piech et al., 2015) to simulate the dynamic changes in learners' knowledge states. Initial logs from these datasets are used to simulate the learner's starting state (Li et al., 2023b). It should be noted that, for previous RL models, the learning items include only knowledge concepts (excluding exercises), whereas in our approach, the exact learning items are set as "exercises" to better align with real-world applications. For instance, while a single concept can have many exercises, prior models only recommended the concept itself and overlooked these exercises. By using exercises directly, our method enables a more detailed and fine-grained learning process for students. In our experimental setup, we simply replace the DKT input from "concepts" to "exercises", ensuring a fair evaluation.

5.3 Baselines

In the Learning Path Recommendation (LPR) task, we compare our approach against existing methods as baselines. It is important to note that all these baseline models only use the sequence of question(or knowledge concept) IDs by learners while ignoring the textual information of the questions and knowledge concepts. Consistent with prior studies (Li et al., 2023b), we use the improvement E_p (Eq. 1) provided by the simulators to evaluate the following methods: (1) KNN (Cover and Hart, 1967), which identifies similar learners based on static learning paths from the training set and recommends the next item, though this often leads to suboptimal performance. (2) GRU4Rec (Hidasi et al., 2015) takes session sequences as input to predict the most likely next learning items through a probability distribution. (3) DQN (Chen et al., 2018) uses a neural network to evaluate and recommend actions with the highest value. Actor-Critic (Konda and Tsitsiklis, 1999) incorporates a GRU encoder into a standard actor-critic framework for recommendations. (4) Contextual Bandits (CB) (Intayoad et al., 2020) frame learning path recommendations as a contextual bandit problem. (5) RLTutor (Kubotani et al., 2021) integrates model-based reinforcement learning with DAS3H (Dwivedi et al., 2018) for adaptive tutoring. (6) CSEAL (Liu et al., 2019) uses an actor-critic framework with cognitive navigation for learning path recommendations. (7) GEHRL (Li et al., 2023b) applies hierarchical reinforcement learning for efficient goal planning. (8) DLPR (Zhang et al., 2024) uses difficulty-driven reinforcement learning to facilitate learning paths. (9) GEPKSD (Li et al., 2024b) leverages privileged knowledge distillation and knowledge graph integration, enabling the RL to adapt learners.

[1] https://pslcdatashop.web.cmu.edu
[2] https://sites.google.com/site/assistmentsdata/home

	Junyi	ASSIST09	TextLog
KNN	0.1343	-0.0932	0.0085
GRU4Rec	0.0993	-0.1344	-0.0002
DQN	0.1536	-0.0267	-
Actor-Critic	0.1916	0.0676	-
CB	0.2098	0.0038	-
RLTutor	-0.1034	0.0784	-
CSEAL	0.2505	0.1009	-
GEHRL	0.4206	0.1971	-
DLPR	0.4972	0.3283	-
GEPKSD	0.3309	**0.5857**	-
ReAL	**0.5724**	0.4304	**0.4847**

Table 2: Performance comparison for learning path recommendation methods at 20 learning steps. The best results are **bold** and the underline means the second best. It should be noted that "-" in the table indicates that the method cannot achieve absolute promotion and meet the learning goals.

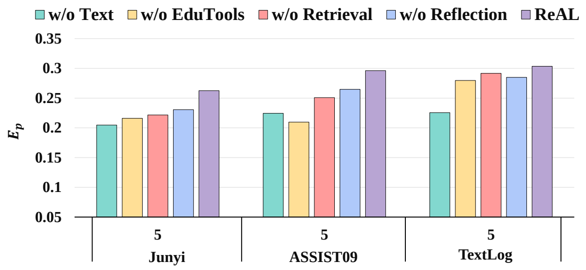

Figure 3: Results of ablation experiments on all datasets.

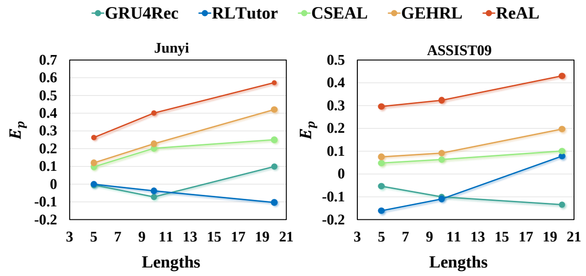

Figure 4: Impact of different lengths on Junyi and ASSIST09 datasets with representative baseline models and our ReAL.

5.4 Implemetation Details

In our framework, we employ three LLM-based models for testing: Llama2-7B, Llama3-8B[3], and the GPT-3.5-turbo provided by OpenAI. The temperature parameter is set to 0.9. The dataset divided method is following (Liu et al., 2019). In particular, our ReAL uses the training dataset to train the simulator and initialize the learner's profile. Then we use the test dataset to achieve inference. More details refer to Appendix and our code is available at https://github.com/karin0018/ReAL.

5.5 Main Results

Table 2 presents the average E_p values of all models across the three datasets, and the learning step is 20. The results reveal several important insights.

- **Performance Comparison**: Our method outperforms all baselines across the Junyi and TextLog datasets, and achieves competitive performance with the state-of-the-art method on the ASSIST09 dataset. These findings highlight the importance of incorporating text information into adaptive learning. It also showcases the potential of large language models in the educational domain. Most methods achieve their best results on the Junyi dataset because it contains fewer knowledge concepts and a smaller exercise set, which narrows the action space and simplifies the decision-making process. Although GEPKSD outperforms on the ASSIST09 dataset, its performance is limited on real-world datasets with sparse interactions due to the RL-based algorithm. This indicates that methods relying heavily on dense feedback signals may lack generalization ability when applied to more challenging environments.

- **Stability Analysis**: While reinforcement learning methods like CSEAL and GEHRL excel with abundant interaction data—benefiting from interactive feedback and long-term rewards—they struggle in sparse data scenarios. For example, in the TextLog dataset, where students have on average only 13 interaction records, these methods fail to train effectively and are unable to generate valid learning path recommendations at the testing stage. Our content-based ReAL framework remains unaffected by sparse data, enabling effective and stable learning path generation.

5.6 Ablation Study

5.6.1 Impact of different modules

We conducted ablation experiments on the key modules of ReAL, and the results are shown in Figure 3, which presents the E_p scores across three datasets when the learning steps are set to 5, and the LLM is using GPT-3.5-turbo. "w/o Text" indicates that we only use the item ID during recommendation

[3]https://www.llama.com/

process, instead of text content for all items (including knowledge concepts and exercises); "w/o EduTools" notes we drop the Educational Tools module, which means we are unable to obtain prior knowledge through additional educational tools, such as the relationships between knowledge concepts. "w/o Retrieval" notes we do not use the previous learners' information, and use the planner to recommend exercise straightly. "w/o Reflection" indicates that we do not update the recommendation strategy or the student's profile during the learning path recommendation process. The experimental results show a significant performance decrease when text information is not utilized in our framework, highlighting the importance of textual content. Additionally, the prior knowledge provided by Educational Tools is also crucial, demonstrating that relying solely on the internal knowledge of the large language model is insufficient; supplementary knowledge is necessary to assist the model in making decisions. The retrieval module in Actor is also important, which demonstrates our human-like decision-making module is useful. Furthermore, since learners' abilities are constantly evolving, the recommendation effectiveness declines when the model no longer dynamically updates the recommendation strategy.

5.6.2 Impact of different path lengths

Figure 4 shows the performance of paths of various lengths generated by different models under three datasets. We compare several representative baseline models with our ReAL. First, the results demonstrate that our method consistently outperforms the baselines across different scenarios, further proving its effectiveness. As the number of learning steps increases, most methods show improved learning outcomes with longer paths, which aligns with educational intuition. Additionally, in scenarios with sufficient training data (e.g.,Junyi and ASSIST09), reinforcement learning-based methods like CSEAL and GEHRL exhibit competitive performance, while RLTutor begins to show instability. Probabilistic prediction methods like GRU4Rec also prove to be less effective. However, these methods rely on abundant training data and fail to converge on the sparse TextLog dataset, making it impossible to reach learning goals during testing (i.e., they fail to run successfully).

Steps	5 Steps	10 Steps	20 Steps
Llama2-7b	0.2782±0.016	0.4057±0.017	0.4121±0.093
Llama3-8b	0.2963±0.048	0.4081±0.012	0.4355±0.042
GPT-3.5-turbo	**0.3035±0.013**	**0.4411±0.015**	**0.4847±0.079**

Table 3: Robustness estimation across different LLMs for TextLog dataset. ± means standard deviation.

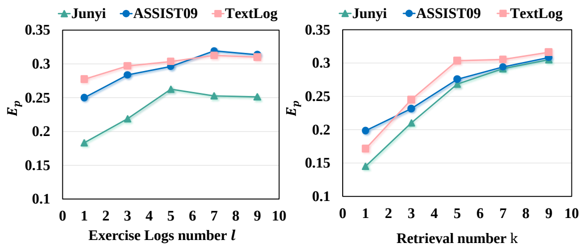

Figure 5: Effects of different exercise logs number l and retrieval number k on three datasets with our ReAL.

5.7 Effects of Different LLMs

To validate the robustness of the ReAL framework across various base models, we conduct experiments using different LLM backbones: Llama-2-7B and Llama-3-8B. The results, presented in Table 3, show that our ReAL framework performs consistently well across these LLMs, demonstrating its robustness. We repeated the experiments three times and calculated the standard variance. The results indicate stable performance within our framework. Notably, the recommendation effectiveness of the Llama-3-8B and GPT-3.5-turbo models surpasses that of Llama-2-7B, suggesting a positive correlation between our framework's performance and the knowledge embedded in larger language models with stronger capabilities.

5.8 Effects of Log number and Top-k number

In the Planner module, LLMs are sensitive to the content of prompts. We analyzed the impact of different exercise log numbers, and the results are shown in Figure 5. The findings indicate that as the number of logs used for inference increases, the accuracy of recommendations first rises and then plateaus, suggesting that LLMs' reasoning ability is still limited by context length. Additionally, we studied the effect of the number of similar learners k used in Actor with fixed candidate number $n = 5$. We found that ReAL's performance also depends on the number of nearest learners. Although the model's performance improves as k increases, the results tend to converge when k exceeds 7. To balance efficiency and effectiveness, we set k to 5.

6 Conclusion

In this paper, we introduced the Retrieval-enhanced Agent for Adaptive Learning (ReAL) framework, which addresses key limitations in existing adaptive learning methods. Prior approaches have largely focused on learner-item interactions based on ID sequences, failing to fully utilize both learner and item information, especially the semantic content of items. Additionally, data-driven reinforcement learning methods struggle in scenarios with sparse learning logs. ReAL, powered by large language models (LLMs), simulates teacher decision-making from both Internal and External perspectives. Internally, ReAL leverages LLMs' semantic understanding and prior knowledge to analyze item texts and learner profiles, generating item candidates. Externally, it retrieves similar learners to select the most suitable item. Moreover, the recommendation strategy is continuously refined through learner feedback using a reflection mechanism. By moving beyond item indexing and relying on semantic content, ReAL demonstrates robust and superior performance across three real-world datasets, particularly in sparse data conditions.

Limitations

Our work is an early attempt to leverage large language models (LLMs) for adaptive learning, and naturally, it comes with several limitations. The first challenge is the high inference cost of LLMs, which makes large-scale deployment in real classroom environments difficult. What's more, while external resources can enrich the recommendation process, the utilization is often inefficient, and the generated outputs may still contain hallucinations. This reflects a broader challenge of LLMs in balancing knowledge grounding with reliable reasoning. Another limitation lies in the evaluation methodology. Following previous works, we adopt a standard evaluation metric widely used in adaptive learning algorithms. This approach has some discrepancies and limitations, and conducting live experiments is expensive. Exploring more effective evaluation algorithms is indeed a valuable direction for future research. Despite these issues, our work shows that LLMs have great potential to solve complex problems in education. In the future, we hope to explore more applications, like using LLMs to help students with specific learning difficulties. We also plan to look more into educational fairness, which is an important issue when applying LLMs.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (2406303), Anhui Provincial Natural Science Foundation (No. 2308085QF229), Anhui Province Science and Technology Innovation Project (202423k09020010), the Fundamental Research Funds for the Central Universities (No. WK2150110034).

References

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Xianyu Chen, Jian Shen, Wei Xia, Jiarui Jin, Yakun Song, Weinan Zhang, Weiwen Liu, Menghui Zhu, Ruiming Tang, Kai Dong, et al. 2023. Set-to-sequence ranking-based concept-aware learning path recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5027–5035.

Yunxiao Chen, Xiaoou Li, Jingchen Liu, and Zhiliang Ying. 2018. Recommendation system for adaptive learning. *Applied psychological measurement*, 42(1):24–41.

Albert Corbett. 2001. Cognitive computer tutors: Solving the two-sigma problem. In *International Conference on User Modeling*, pages 137–147. Springer.

Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Pragya Dwivedi, Vibhor Kant, and Kamal K Bharadwaj. 2018. Learning path recommendation based on modified variable length genetic algorithm. *Education and information technologies*, 23:819–836.

Lumbardh Elshani and Krenare Pireva Nuçi. 2021. Constructing a personalized learning path using genetic algorithms approach. *arXiv preprint arXiv:2104.11276*.

Aoran Gan, Hao Yu, Kai Zhang, Qi Liu, Wenyu Yan, Zhenya Huang, Shiwei Tong, and Guoping Hu. 2025. Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey. *Preprint*, arXiv:2504.14891.

Weibo Gao, Qi Liu, Zhenya Huang, Yu Yin, Haoyang Bi, Mu-Chun Wang, Jianhui Ma, Shijin Wang, and Yu Su. 2021. Rcd: Relation map driven cognitive diagnosis for intelligent education systems. SIGIR '21, page 501–510, New York, NY, USA. Association for Computing Machinery.

Weibo Gao, Qi Liu, Linan Yue, Fangzhou Yao, Rui Lv, Zheng Zhang, Hao Wang, and Zhenya Huang. 2025. Agent4edu: generating learner response data by generative agents for intelligent education systems. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25. AAAI Press.

Senyu Han, Lu Chen, Li-Min Lin, Zhengshan Xu, and Kai Yu. 2024. IBSEN: Director-actor agent collaboration for controllable and interactive drama script generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1607–1619, Bangkok, Thailand. Association for Computational Linguistics.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. 2019. Exploring multi-objective exercise recommendations in online education systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1261–1270.

Wacharawan Intayoad, Chayapol Kamyod, and Punnarumol Temdee. 2020. Reinforcement learning based on contextual bandits for personalized online learning recommendation systems. *Wireless Personal Communications*, 115(4):2917–2932.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Yoshiki Kubotani, Yoshihiro Fukuhara, and Shigeo Morishima. 2021. Rltutor: Reinforcement learning based adaptive tutoring system by modeling virtual student with fewer interactions. *arXiv preprint arXiv:2108.00268*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Qingyao Li, Lingyue Fu, Weiming Zhang, Xianyu Chen, Jingwei Yu, Wei Xia, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023a. Adapting large language models for education: Foundational capabilities, potentials, and challenges. *arXiv preprint arXiv:2401.08664*.

Qingyao Li, Wei Xia, Kounianhua Du, Qiji Zhang, Weinan Zhang, Ruiming Tang, and Yong Yu. 2024a. Learning structure and knowledge aware representation with large language models for concept recommendation. *arXiv preprint arXiv:2405.12442*.

Qingyao Li, Wei Xia, Li'ang Yin, Jiarui Jin, and Yong Yu. 2024b. Privileged knowledge state distillation for reinforcement learning-based educational path recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 1621–1630. Association for Computing Machinery.

Qingyao Li, Wei Xia, Li'ang Yin, Jian Shen, Renting Rui, Weinan Zhang, Xianyu Chen, Ruiming Tang, and Yong Yu. 2023b. Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, page 1318–1327.

Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 627–635. Association for Computing Machinery.

Amir Hossein Nabizadeh, Daniel Goncalves, Sandra Gama, Joaquim Jorge, and Hamed N Rafsanjani. 2020a. Adaptive learning path recommender approach using auxiliary learning objects. *Computers & Education*, 147:103777.

Amir Hossein Nabizadeh, José Paulo Leal, Hamed N Rafsanjani, and Rajiv Ratn Shah. 2020b. Learning path personalization and recommendation methods: A survey of the state-of-the-art. *Expert Systems with Applications*, 159:113596.

Amir Hossein Nabizadeh, Alípio Mário Jorge, and José Paulo Leal. 2017. Rutico: Recommending successful learning paths under time constraints. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization*, pages 153–158.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

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

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Large language models are learnable planners for long-term recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1893–1903. Association for Computing Machinery.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Jinze Wu, Haotian Zhang, Zhenya Huang, Liang Ding, Qi Liu, Jing Sha, Enhong Chen, and Shijin Wang. 2024. Graph-based student knowledge profile for online intelligent education. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 379–387. SIAM.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Songlin Xu, Xinyu Zhang, and Lianhui Qin. 2024. Eduagent: Generative student agents in learning. *arXiv preprint arXiv:2404.07963*.

Qisen Yang, Zekun Wang, Honghui Chen, Shenzhi Wang, Yifan Pu, Xin Gao, Wenhao Huang, Shiji Song, and Gao Huang. 2024. PsychoGAT: A novel psychological measurement paradigm through interactive fiction games with LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14470–14505, Bangkok, Thailand. Association for Computational Linguistics.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2025. Evaluation of retrieval-augmented generation: A survey. In *Big Data*, pages 102–120, Singapore. Springer Nature Singapore.

Haotian Zhang, Shuanghong Shen, Bihan Xu, Zhenya Huang, Jinze Wu, Jing Sha, and Shijin Wang. 2024. Item-difficulty-aware learning path recommendation: From a real walking perspective. KDD '24, page 4167–4178. Association for Computing Machinery.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

exercise
In parallelogram ABCD, diagonals AC and BD intersect at point O. Line EF passes through point O, intersecting AD and BC at points E and F, respectively. Prove that OE = OF.
knowledge
Properties of a Parallelogram.

Table 4: An example of TextLog dataset.

A Appendix

A.1 Educational Tools

We follow the structure proposed by the original work of Deep Knowledge Tracing (DKT) (Piech et al., 2015), and extend an embedding layer that reduces the large feature space (Liu et al., 2019). We use the 128-dimensional embedding layer and use Adam (Kingma and Ba, 2014) as our optimizer and the learning rate is set to be 0.001.

We constructed the knowledge transition graphs from exercise data using the statistical methods described in the RCD (Gao et al., 2021) paper. This tool is open-source and available at `https://github.com/bigdata-ustc/RCD`. During the inference process, we employ the Cognitive Navigation algorithm from CSEAL (Liu et al., 2019), which helps maintain the logical consistency of the recommended learning paths (e.g., avoiding the recommendation of calculus to junior students) and reduces the search space.

After selecting the knowledge concept to be learned, we add the relevant exercises to the initial set. Due to the context length limitations of large language models, we limit the set size to 20. If > 20, we rank and filter out the already mastered knowledge concepts and exercises based on the mastery level provided by DKT.

A.2 An example of TextLog dataset

We present a data sample from the TextLog dataset in Table 4. Unlike Junyi and ASSIST, which only have the name of the knowledge concept and limited information for exercises, TextLog have complete textual information.

A.3 More Details

We set a higher temperature to encourage the LLM to generate diverse outputs, enabling more personalized student profiles and candidate exercise sets.

Temperature	0.1	0.3	0.5	0.7	0.9
TextLog	0.2542	0.2638	0.2829	0.3244	0.3035

Table 5: Performance of TextLog under different temperature settings with step=5.

We conducted experiments on the TextLog dataset with step=5 under different temperature settings. The results are shown on Table 5. The results show that a higher temperature improves recommendation performance. We believe this is because our framework relies on the LLM to generate student profiles and candidate exercises adaptively. A lower temperature produces more uniform outputs, such as similar student profiles, limiting personalization and reducing learning path effectiveness.

A.4 Case Study

Table 6 presents detailed examples of our input prompt template alongside the corresponding output content generated for the TextLog dataset.We demonstrate that large language models (LLMs) can infer various aspects of a learner's abilities and learning preferences. Specifically, the models can identify the learner's strengths in particular knowledge concepts based on their previous exercise records. In addition, ReAL further enhances the recommendation process by leveraging the behavior and performance of similar learners who have faced comparable challenges. This ensures that the recommended exercises are relevant and effective in helping the learner make progress. The results from the TextLog demonstrate that the exercises recommended by ReAL significantly contribute to the learner mastering the target knowledge concepts, highlighting the practical effectiveness of the approach in real-world learning scenarios.

step = 0, MAXSTEP = 5, $E_t^{g_i}$ = 0.51827 (note as the learning mastery level of the learning goal)

Input: Prompts of learning profile $\mathrm{P_L}$

You're a seasoned math teacher with multiple years of teaching experience. Please use one sentence to summarize the student's learning ability and learning preferences from the following learning log {1. Question: The equation of the line with the maximum distance from the origin that passes through the point (12, 0). Answer: correct. 2. Question: For triangle ABC, the coordinates of vertex B are (-1, -3). The equation of the line containing the height from AB, denoted as CE, is $4x + 3y - 7 = 0$. The equation of the median AD from vertex B to side BC is $x - 3y - 3 = 0$. Find the coordinates of point C and the equation of line AB. Answer: wrong. 3. Question: The equation of the line passing through point P(2, 3) and having intercepts on the coordinate axes that are opposite in sign. Answer: correct. 4. Question: Which of the following equations can represent a circle? Answer: correct }. The learning goal is: {The coordinate characteristics of points on the graph of a linear function}.

Output: Learner's Profile

student-learning-ability: Based on this learning log, the student demonstrates a mixed ability in math, showing strengths in solving linear and circular equations, but struggling with finding the equation of a line passing through two points and identifying the equation of a line given its midpoint and slope., student-learning-preference: Based on this learning log, it appears that the student prefers to learn and apply mathematical concepts through direct and explicit instruction, as they tend to answer questions correctly when the concepts are presented in a straightforward and concrete manner, but struggle with questions that require more complex problem-solving and application.

Input: Prompts of Reflection $\mathrm{P_R}$

Please use one sentence to reflect the strategy of recommend questions from the learner's feedback:[...].

Output: Recommendation strategy

To effectively recommend questions, I would focus on identifying patterns and relationships between the questions, such as the types of geometry problems, the level of difficulty, and the types of concepts involved, and then use this information to suggest questions that are likely to be relevant and challenging for the student.

Input: The Prompts for the exercise recommendation $\mathrm{P_C}$

You're a seasoned math teacher with ten years of teaching experience. Given the following history text: [question: 'The equation of the line that passes through point (1,2) and the origin with the maximum distance', 'answer': 'correct', , question: 'In triangle ABC, vertex B(-1,-3), the equation of the line of the altitude CE is 4x + 3y - 7 = 0. The equation of the median AD on side BC is x - 3y - 3 = 0. The coordinates of point C and the equation of line AB', 'answer': 'wrong', , ...] and the recommended reflection: [...], the student profile: [...], the knowledge learning goal: The coordinates of points on the graph of a linear function. Here are the candidate question list: [question-id: '5244', 'question-text': 'Point (m, m+3) on the function y=-x+2, find m', question-id: '19', 'question-text': 'Three points (14, 27), (a, 10) are on the same line, find a', ..., question-id: '6291', 'question-text': 'For the line y = 1/2x - 3/2, find the value of 4y - 2x + 3', ...]. Please provide the 5 suitable questions from the above list that can help the student achieve the learning goal efficiently. Furthermore, you should give your reason for recommending these questions. For example, the output format should be: [question-id-list: [the 5 question-ids that you recommend], 'recommend-reason': ['recommend reason']], except this format, please do not output anything.

Table 6: An example of the recommendation processing on TextLog dataset, including Planner and Reflector.

Output: Recommendation exercises, explanation and prediction
{['question-id-list': [4547, 5454, 473, 7972, 5244],'recommend-reason': ['These questions are directly related to the learning goal of identifying the coordinate features of points on the graph of a linear function. Question 4547 and 5454 are about finding the equation of a line passing through two points, which is a fundamental concept in linear functions. Question 473 and 7972 are about the graph of a linear function, including its intercepts and slope, which is crucial for understanding the coordinate features of points on the graph. Question 5244 is about finding the point on the graph of a linear function given its coordinates, which is a common scenario in real-world problems. These questions are also well-suited for a student with a mixed ability in math, as they are challenging but not overwhelming.]}
Input: Recommendation exercises, previous learners.
load the training learners state from ../data/memory.pt... retrieve learners who have similar learning state with current learner...
Output: Recommend question-id: 7972
step = 1, MAXSTEP = 5, $E^{g_i}_{t+1}$ = 0.5791 (note as the learning mastery level of the learning goal) ...

Table 7: An example of the learning path recommendation processing on TextLog dataset, including Actor.