Learning from Long-Term Engagement: Adaptive Tutoring Dialogue Planning for Personalized Education

Zhiang Dong, Zhenlong Dai, Xiangwei Lv, Jingyuan Chen*

Zhejiang University

{dongza,zhenlongdai,xiangwei.lv,jingyuanchen}@zju.edu.cn

Abstract

With the advancements of large language models (LLMs), intelligent tutoring systems have witnessed significant progress. The extensive knowledge and reasoning capabilities of LLMs enable intelligent tutoring systems to generate more helpful tutoring dialogues with scaffolding instructions. However, these systems fail to provide scaffolds that align with the personalized needs of students due to the lack of attention to the long-term learning process of students. Meanwhile, the pursuit of more suitable scaffolds through complex reasoning may result in additional computational overhead. To address these issues, we propose LEAP, a **L**ong-term **E**ducational **A**daptive **P**lanning system that can model students' long-term learning process. Specifically, LEAP plans for scaffolds through collaboration of direct planning and thoughtful reasoning to improve efficiency and captures students' long-term learning progress through cognitive state extraction. Then we propose LEAD, a **L**ong-term **E**ducational **A**rchive **D**ataset to alleviate the lack of data and validate the effectiveness of LEAP, which is constructed through real-world students' reactions and simulation of the teacher-student interactions. Experiments on several datasets demonstrate the effectiveness of LEAP.

Code & Datasets — https://github.com/PlayerDza/LEAP

Introduction

Recent advancements in large language models (LLMs) have significantly enhanced the sophistication, accessibility, and adaptability of intelligent education systems. These models now play a pivotal role in facilitating personalized learning, while simultaneously helping to reduce disparities in access to educational resources (Liu et al. 2024a; Wu et al. 2025).

Early approaches to LLM-based tutoring primarily relied on simulating human teachers through direct prompt-based interaction, leveraging the LLM to deliver personalized dialogue and instructional guidance (Hirunyasiri et al. 2023; Phung et al. 2023). However, these methods are constrained by the representational and inferential capacities of LLMs, as well as the specificity and effectiveness of the prompts.

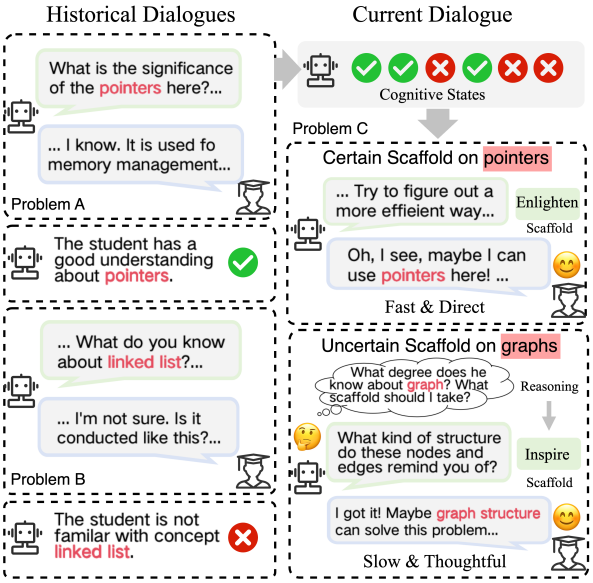

Figure 1: An illustration of the long-term tutoring dialogue process and the planning for appropriate guidance.

Consequently, they often fall short in providing targeted and adaptive support tailored to individual student needs.

Recent developments in LLM-based tutoring dialogue systems have begun to address these limitations by capitalizing on the rich domain knowledge and advanced reasoning abilities of LLMs. These systems can deliver individualized instructional support that adapts to students' evolving knowledge states and proficiency levels (Zhang et al. 2024b; Dan et al. 2023). Drawing inspiration from experienced human teachers, advanced systems have started to incorporate dynamic scaffolding instructions, teaching strategies designed to provide incremental support and facilitate deeper understanding of targeted concepts (Kong et al. 2024; Huang et al. 2025). Such scaffolding can significantly enhance students' learning efficiency and engagement.

However, there are still gaps between tutoring dialogue systems and experienced human teachers. Responsive Teaching Theory (Robertson et al. 2015) emphasizes that truly personalized scaffolding requires continuous monitoring and assessment of students' long-term learning trajecto-

*Corresponding Author.

ries—an aspect that extant dialogue planning methods insufficiently address (Liu et al. 2024a). Moreover, aligning scaffolding decisions tightly with a learner's current cognitive state necessitates sophisticated reasoning processes; these, in turn, can incur prohibitive computational overhead and hinder real-time responsiveness.

To bridge these gaps, we propose LEAP, a **L**ong-term **E**ducational **A**daptive **P**lanning system to fully utilize the long-term learning process of students. As illustrated in Figure 1, the scaffolding instruction planning module improves efficiency through the collaboration between direct planning and reasoning through Monte Carlo Tree Search (Väth, Vanderlyn, and Vu 2023). We use a cognitive state extraction module to obtain the long-term cognitive state, enabling efficient utilization through memory storage and retrieval mechanisms. The scaffolding instruction planning module we employed includes a direct planner and an MCTS planner, where the direct planner provides fast and direct planning, while the MCTS planner offers deep and thoughtful planning. It uses confidence to decide whether MCTS reasoning is needed to enhance accuracy. To enhance the model's planning capability, we adopt a two-stage training method for the direct planner, consisting of offline reinforcement learning training and MCTS simulation training.

To support the development and evaluation of LEAP, we propose LEAD, a **L**ong-term **E**ducational **A**rchive **D**ataset built from real-world student interactions on an online educational platform. LEAD constructs the student's learning process by identifying and analyzing the differences in their multi-round submission code. We then utilize the "Dean-Teacher-Student" (Liu et al. 2024a) framework to simulate student-teacher interactions, ensuring that the generated dialogues align with the learning process demonstrated in the student's multi-round submissions. Additionally, we incorporated scaffolding theory (Wiske 1998) to implement the scaffolding instructions.

The main contributions of this work are:

- We introduce LEAP, a framework mainly focusing on planning scaffold in the long-term learning process of students. It includes cognitive state extraction and scaffolding instruction modules.
- We construct LEAD, a long-term tutoring dialogue dataset based on real-world students' reactions across diverse problems and courses.
- Experiments on several datasets demonstrate the effectiveness of our framework.

Related Work

Dialogue Tutoring Datasets

Building efficient tutoring systems requires high-quality data resources that closely reflect real-world teaching scenarios. Early attempts at collecting and constructing dialogue tutoring datasets mainly focus on manual creation (Nye, Graesser, and Hu 2014; Stasaski, Kao, and Hearst 2020) and field collection (Howe et al. 2019; Caines et al. 2020). However, these methods may face challenges such as high annotation costs and limited scalability. Recently, with the support of LLMs, LLM-driven synthetic data generation became a possible solution (Long et al. 2024). Simulating teaching and learning processes to construct education resources is a typical application of LLMs (Yue et al. 2024; Xu, Zhang, and Qin 2024). For dialogue tutoring datasets that simulate teacher-student interactions, LLMs are primarily used to replicate specific instructional scenarios with varying knowledge backgrounds (Liu et al. 2024b; Macina et al. 2023a; Chen et al. 2024) or teaching methods (Kwon et al. 2024; Macina et al. 2023b), such as Socratic teaching (Liu et al. 2024a).

However, few existing methods focus on students' long-term learning process performance, specifically the feedback of the same student across different problems.

LLM-enhanced Tutoring Systems

With the development of LLMs, intelligent tutoring systems are experiencing significant advancements. (Park et al. 2024; Hirunyasiri et al. 2023; Dai et al. 2025; Lv et al. 2025a) prompt GPT to build dialogue tutoring systems. Meanwhile, the fine-tuning approach endows LLMs with human-like teaching capabilities by integrating domain knowledge and instructional methods (Dan et al. 2023; Liu et al. 2024a; Dai et al. 2024). However, LLMs struggle to fulfill the role of a teacher in educational settings effectively (Abdelghani et al. 2024; Qiu et al. 2025), as they may diagnose students' learning status incorrectly and prematurely reveal answers. SocraticLM (Liu et al. 2024a) and MathDial (Macina et al. 2023a) leverage Socratic teaching and scaffolding to mitigate the prematurely revealing of answers and provide guidance step-by-step. Efforts are also being made to use LLM-powered agents to simulate the processes of teaching and learning (Gao et al. 2025; Zhang et al. 2024b; Lv et al. 2025b).

These models explore providing personalized tutoring for students from various perspectives. However, they fail to effectively leverage the long-term tutoring dialogues in the student's learning process.

Dataset Construction

Data Collection

Due to the lack of attention to students' long-term learning process across different problems and courses, dialogue tutoring datasets that can effectively describe this learning process are scarce. To address this limitation, we propose LEAD. Our data source is PTADisc (Hu et al. 2023), a dataset that documents students' authentic multi-round submissions on the online educational platform PTA[1]. We analyze the differences in multi-round submissions of programming problems to understand the students' learning process. Based on this analysis, we simulate teacher-student interactions and use these interactions to construct a tutoring dialogue dataset. Since only some courses include programming problems, we selected three courses with representative differences in difficulty levels, **Data Structures and Algorithms Analysis** (denoted as Data Structure), **C++**

[1]https://pintia.cn/

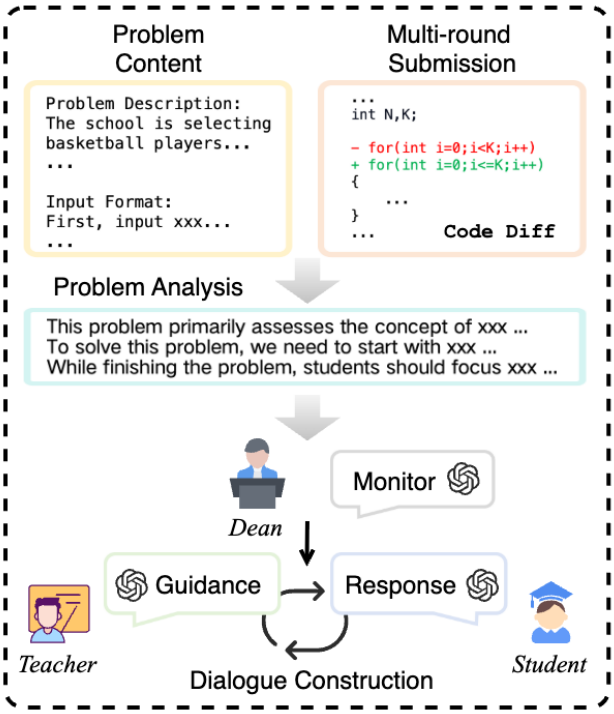

Figure 2: The construction process of LEAD dataset.

Programming, and **Multivariate Statistical Analysis** (denoted as Statistical Analysis). Detailed statistics of the three courses in LEAD are described in Appendix.

Dataset Construction Process

In order to simulate the tutoring process, we follow the "Dean-Teacher-Student" structure (Liu et al. 2024a) to propose a multi-agent framework to formulate the entire dialogue, as shown in Figure 2. The multi-agent framework contains three agents: *Dean*, *Teacher*, and *Student*. Additionally, we employ a teaching strategy in the tutoring process. To ensure the quality of the dataset, we conducted a quality evaluation after the construction process. We employ GPT-4o to implement these agents.

Construction Process. In the construction process, we first utilize the problem content and multi-round submission to generate the problem analysis to detail the problem and analyze the problem-solving process. The multi-round submission is processed through `Code Diff`[2] to identify the difference between multi-round codes in the programming problems. Code differences reflect the changes in a student's understanding of the problem and learning state throughout multiple submissions. Centering on this learning process, we simulate the student's learning progress by analyzing the differences between each submission. Then, we utilize the problem analysis, problem content, and multi-round submission to construct the dialogue. The dialogue construction process is implemented through a multi-agent framework to simulate teacher and student. Specifically, we employ the *Teacher* agent and *Student* agent to reconstruct the interactions. By leveraging problem analysis and code differences, we simulate how the *Teacher* should guide the *Student* to help him identify issues and make necessary modifications to his code. This guidance should adopt a scaffolding strategy. Then the *Student* responds to the guidance to formulate the dialogues. During this tutoring process, the *Dean* agent monitors the entire dialogue to ensure that the *Teacher*'s guidance aligns with the context.

[2]An approach that can identify the difference between two codes.

	DS	CPP	SA
#Students	1336	956	169
#Problems	352	343	26
#Dialogues	34,636	25,863	4,991
#Difficulty	0.46	0.58	0.36
#Avg. Problems	4.79	4.98	5.21
#Avg. $\text{Words}_{\text{Teacher}}$	66.98	66.79	62.70
#Avg. $\text{Words}_{\text{Student}}$	45.38	39.07	42.75
Reconstruction	0.902	0.931	0.926

Table 1: Statistics of the proposed LEAD dataset, including three courses: data structure (DS), C++ programming (CPP), and statistical analysis (SA). Reconstruction is evaluated through CodeBertScore.

Scaffolding Strategy. Inspired by the Scaffolding Theory (Wiske 1998), we categorized the teacher's guidance into four scaffolding instructions: **Inspire**, **Introduce**, **Enlighten**, and **Summarize**. During the generation of the teacher's guidance, each scaffolding instruction is explicitly labeled accordingly. Detailed classifications and examples of these instructions can be found in Appendix.

Quality Evaluation. After the construction process, we conducted both manual and automatic evaluations to verify the authenticity and accuracy of LEAD, as well as its alignment with the context of tutorial dialogues. The manual evaluation focused on whether dialogues revolved around code differences and aligned with the teaching scenario. For automatic evaluation, we assess dialogue quality by reconstructing the code based on the generated dialogues. We supply the student's prior code, use the generated dialogue to reconstruct the next round, and compare it with the actual submission for similarity. Detailed quality evaluation process can be found in Appendix.

Dataset Statistics

The statistics of the proposed LEAD are described in Table 1. Note that here we use student score rates to reflect the difficulty of the course. Construction denotes the automatic evaluation of dialogue quality mentioned above, and we utilize CodeBertScore as metrics.

Methodology

Task Definition

The goal of the multi-session student tutoring dialogue task is to utilize the dialogue context $\mathcal{C}$ and the historical tutor-

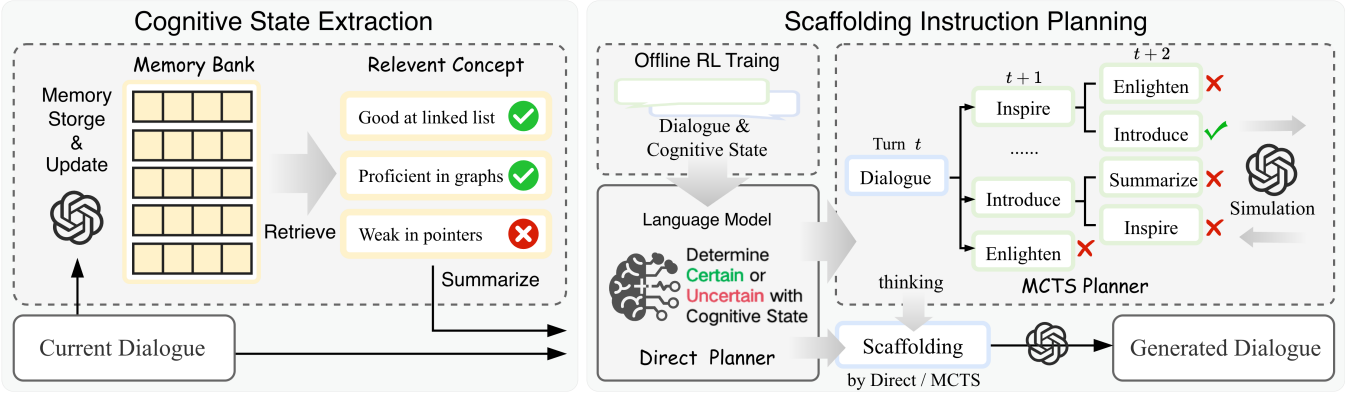

Figure 3: Framework overview. The cognitive state extraction module and the scaffolding instruction planning module extract the student's long-term cognitive states and planning for the teaching scaffolds.

ing dialogues $\mathcal{H}$ to generate guidance d_g, which aligns with the principles of Socratic teaching. The current dialogue $\mathcal{C}$ is defined as $\{d_1, d_2, ..., d_t\}$, where t denotes the t-th turn and $d = \{d_r, d_g\}$ denotes the t-th dialogue, d_r and d_g denote the response of student and guidance of teacher. Historical dialogues $\mathcal{H}$ is defined as $\{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{n-1}\}$, where n denotes the total n-th session before the current dialogue and the current dialogue is $\mathcal{C}_n$.

Following previous works (Wang et al. 2020; Deng et al. 2024), the entire tutoring dialogue task can be formulated as a Markov Decision Process (MDP). At each turn t, according to the observation on the dialogue context $\mathcal{C}$ and dialogue history $\mathcal{H}$, the tutoring dialogue system chooses an action $a \in \mathcal{A}$, where $\mathcal{A}$ denotes a pre-defined strategy set, which is the scaffolding instructions. The objective is to learn a policy π maximizing the expected cumulative rewards over the observed dialogue context $\mathcal{C}$ and dialogue history $\mathcal{H}$ as:

$$\pi^* = \arg\max_{\pi_\theta} \left[\sum_{t=0}^{T} r\left(s_t, a_t\right) \right], \tag{1}$$

where $r(\cdot)$ denotes the reward function, denoted as r_t, s_t denotes the state of the dialogue. With the policy, we can guide the LLM to generate more targeted tutoring dialogue.

Framework Overview

The proposed method includes two modules, **cognitive state extraction** and **scaffolding instruction planning**, as illustrated in Figure 3. Cognitive state extraction leverages the student's grasp of related knowledge concepts from historical dialogues on other problems to assess their understanding of the current problem. Scaffolding instruction planning employs the direct planner and the MCTS planner to implement the planning of scaffolds. We adopt a two-stage training method, offline RL training and MCTS simulation training, to train this module in order to coordinate two different planners.

Cognitive State Extraction

To diagnose students' cognitive states from their responses to diverse problems, inspired by previous paradigms (Zhang et al. 2024a; Zhong et al. 2024; Li et al. 2024), we employ a memory mechanism designed for long-term preservation across dialogue sessions.

Memory Storage & Update. For each historical dialogue, we first generate a core summary of the dialogue. The focus of the summary is to highlight the core aspects of the dialogue while reducing storage costs. This process is accomplished by LLMs as $c = C(u)$, where $u \in a, g$ denotes the teacher or student utterance, $C(\cdot)$ denotes the summary process of LLMs. The summary is encoded with $E(\cdot)$ and stored alongside its timestamp t in the memory bank as: $M = \{(E(c_i), t_i)\}$, where $E(\cdot)$ denotes the text encoder, i denotes the i-th memory in M. This compact representation lowers storage overhead and speeds up subsequent retrieval. Records in M older than a threshold (*e.g.*, 180 days) are removed to maintain memory timeliness. The threshold can be adjusted if record volume is excessive.

Memory Retrieval. During the retrieval process, to enhance the accuracy of retrieving relevant events, we primarily rely on semantic similarity for retrieval. Additionally, since the events reflect the student's level of knowledge mastery, which may diminish over time, we also take into account concept overlap and time decay in the retrieval process. Semantic similarity x_{sem} is obtained by calculating the semantic similarity between the query (current context) and each memory key (encoded summary). Topic overlap x_{over} is defined as:

$$x_{over} = -\frac{1}{2}\left(\log \frac{|L_q \cap L_k|}{|L_q|} + \log \frac{|L_q \cap L_k|}{|L_k|}\right), \tag{2}$$

where L_q and L_k denote the noun libraries of query and key. The time decay weight λ is defined as $\lambda = e^{-\Delta t/\tau}$, where Δt denotes the time gap between query and key. The overall score x can be calculated as $x = \lambda(x_{over} + x_{sem})$. The calculated overall score x is used as the retrieval criterion, and the top-k highest-valued memories is selected. For the final obtained memories, we use LLMs to integrate them, denoted as m.

Scaffolding Instruction Planning

In the scaffolding instruction planning module, we use both a direct planner and an MCTS planner for planning. We employ an LLM both as a critic and to simulate the roles of students and teachers in MCTS.

Direct Planner. For the direct planner, we use a pretrained language model to control the dialogue planning process. The language model is connected to both the policy network $\pi_\theta(a|s)$ and the Q-network $Q_\phi(s,a)$ via two MLPs, for action prediction and state evaluation. We employ an LLM as the reward function to simulate the student's response after receiving guidance, in order to better train the model. The use of $Q_\phi(s,a)$ allows for better utilization of LLM feedback, helping the $\pi_\theta(a|s)$ generate more accurate actions a_t at the dialogue state s_t. The input of the direct planner is the mixture of cognitive state m and dialogue d.

The direct planner is first trained through RL training. For $\pi_\theta(a|s)$ and $Q_\phi(s,a)$, we use the following optimization method:

$$\begin{cases} \mathcal{L}_{actor}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \hat{Q}(s_i,a_i)\log \pi_\theta(a_i|s_i), \\ \mathcal{L}_{critic}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\Big(Q_\phi(s_i,a_i) - \hat{Q}(s_i,a_i)\Big)^2, \\ \mathcal{L} = \mathcal{L}_{actor}(\theta) + \alpha\mathcal{L}_{critic}(\phi), \end{cases} \quad (3)$$

where $\hat{Q}(s_t,a_t)$ denotes the cumulative rewards. $\mathcal{L}$ is the entire loss function and α is the loss weight.

MCTS Planner. Following previous works (Yu, Chen, and Yu 2023; He et al. 2024), we employ MCTS to implement the deep and thoughtful reasoning process. MCTS mainly consists of four processes: iteratively performing action selection, search tree expansion, action evaluation, and backpropagation to update tree statistics. Through this process, MCTS can simulate scaffolding strategies and predict more accurate actions a_t. Meanwhile, we initialize MCTS with the prior probabilities from the direct planner, which gives MCTS a better understanding of the student's knowledge state and facilitates the adoption of more targeted instructional strategies.

In MCTS simulation training, we first simulate teacher-student interactions through MCTS planner, and then use the simulated interaction data to further train the direct planner. For the policy network $\pi_\theta(a|s)$ and the Q-network $Q_\phi(s,a)$, we use the following optimization method:

$$\begin{cases} \mathcal{L}'_{\text{actor}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} A_\phi(s_i,a_i)\log \pi_\theta(a_i|s_i), \\ \mathcal{L}'_{critic}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i^* - Q_\phi(s_i,a_i)\right)^2, \\ \mathcal{L}' = \mathcal{L}'_{actor}(\theta) + \beta\mathcal{L}'_{critic}(\phi), \end{cases} \quad (4)$$

where $A_\phi(s_i,a_i) = Q_\phi(s_i,a_i) - \hat{Q}(s_i,a_i)$ and $y_t^* = r_t + \gamma \max_a Q_\phi(s_{t+q},a)$ denotes one-step temporal-difference target. $\mathcal{L}'$ is the entire loss function and β is the loss weight.

Planner Collaboration. The core of scaffolding instruction planning lies in flexibly invoking two different planners based on students' varying cognitive states for scaffolding planning. If the model has high confidence in the scaffold that should be applied based on the student's current cognitive state, the scaffold selected by the direct planner can be used directly. If the confidence is low, MCTS should be invoked to perform reasoning before selecting the scaffold.

In the scaffolding instruction planning module, we use the action distribution of the direct planner $\pi_\theta(a|s)$ to measure this confidence. Specifically, we calculate the difference between the highest and the second-highest values of $\pi_\theta(a|s)$, denoted as ω. When ω exceeds a threshold ρ, the action is generated directly by the direct planner; otherwise, MCTS is used. The value ω reflects the model's confidence in the student's knowledge state and the corresponding scaffolding strategy for the dialogue. By controlling ρ, we can adjust the proportion of MCTS usage.

Experiments

Experimental Settings

Datasets. We conducted experiments on three courses from the LEAD dataset—Data Structure, C++ Programming, and Statistical Analysis—with predefined scaffolds. Following the multi-session dialogue structure (Xu, Szlam, and Weston 2022), each student's conversation was formatted into sessions, using session 1 as historical context and evaluating the model on sessions 2–4.

Evaluation Metrics. Following previous tutoring dialogue systems, we evaluate the effectiveness of LEAP using both automatic evaluation and LLM evaluation. For automatic evaluation, we assess the quality of generated dialogue through two widely used metrics BLEU (Papineni et al. 2002) and ROUGE (Lin 2004). B denote BLEU and R denote ROUGE. For LLM evaluation, we assess the generated dialogues based on two dimensions: coherence and insight (rating from 0 to 1). Coherence measures the alignment of the guidance with the given context, and insight evaluates whether the generated guidance effectively addresses the student's issues and provides meaningful instructions. In the subsequent tables, **bold** numbers indicate the best performance.

Baseline Methods. To validate the effectiveness of the proposed method, we conduct experiments on several baseline methods. **Direct prompting** directly prompts LLMs to generate guidance based on historical dialogues. **CoT prompting** utilizes chain of thought (Wei et al. 2022) to prompt LLMs generate guidance step-by-step. **ICL prompting** leverage dialogue examples to implement in-context learning (Dong et al. 2022) to generate guidance. Moreover, educational LLMs EduChat (Dan et al. 2023) and the tutoring dialogue generation method **SocraticLM** (Liu et al. 2024a) are also included.

Experimental Setup. We utilize PyTorch to implement both the baseline methods and our proposed LEAP framework. We employ GPT-4o to implement the prompting methods in baselines. We implement the direct planner through RoBERTa-large and the MCTS training through GPT-4o. More details of experimental setup can be found in Appendix.

Methods	Session 2					Session 3					Session 4				
	B-2	B-3	R-L	Coh.	Ins.	B-2	B-3	R-L	Coh.	Ins.	B-2	B-3	R-L	Coh.	Ins.
Data Structure															
Direct Prompting	12.68	4.66	18.85	0.605	0.632	13.04	4.86	19.06	0.624	0.657	12.91	4.83	19.02	0.632	0.663
CoT Prompting	12.83	4.89	19.36	0.624	0.668	13.19	4.89	19.31	0.617	0.672	13.12	4.94	19.29	0.637	0.685
ICL Prompting	13.41	5.32	19.77	0.620	0.685	13.88	5.46	19.82	0.636	0.678	13.92	5.41	19.68	0.654	0.706
EduChat	10.58	4.22	16.55	0.527	0.641	10.51	4.23	16.46	0.538	0.673	10.72	4.42	17.09	0.569	0.694
SocraticLM	11.72	4.46	17.74	0.554	0.715	11.85	4.52	17.88	0.576	0.724	12.06	4.63	18.26	0.585	0.743
LEAP	**14.34**	**5.98**	**20.74**	**0.748**	**0.796**	**14.63**	**6.25**	**21.02**	**0.765**	**0.809**	**14.75**	**6.36**	**21.11**	**0.782**	**0.826**
C++ Programming															
Direct Prompting	12.66	4.72	18.27	0.682	0.633	12.85	4.90	18.52	0.695	0.658	12.73	4.79	18.53	0.709	0.664
CoT Prompting	12.46	4.64	18.62	0.710	0.663	12.59	4.75	18.67	0.703	0.667	12.52	4.76	18.66	0.723	0.680
ICL Prompting	13.09	5.16	19.08	0.724	0.691	13.17	5.26	19.22	0.742	0.684	13.31	5.23	19.35	0.758	0.712
EduChat	10.65	4.24	16.44	0.644	0.655	10.76	4.13	16.49	0.655	0.687	10.71	4.18	16.53	0.686	0.708
SocraticLM	11.69	4.36	17.71	0.673	0.701	11.72	4.43	17.56	0.695	0.713	11.87	4.63	17.82	0.704	0.729
LEAP	**13.64**	**5.82**	**19.73**	**0.815**	**0.774**	**13.76**	**5.97**	**19.89**	**0.824**	**0.786**	**13.95**	**6.04**	**20.06**	**0.847**	**0.805**
Statistical Analysis															
Direct Prompting	12.17	4.21	19.06	0.764	0.644	12.32	4.49	19.31	0.783	0.659	12.44	4.41	19.33	0.791	0.675
CoT Prompting	12.32	4.28	19.15	0.804	0.665	12.41	4.51	19.27	0.797	0.668	12.51	4.53	19.46	0.817	0.682
ICL Prompting	12.65	5.02	19.51	0.797	0.687	12.84	5.26	19.62	0.813	0.68	12.82	5.17	19.84	0.831	0.708
EduChat	10.87	3.84	16.48	0.727	0.684	10.76	4.09	16.76	0.738	0.716	10.82	4.20	16.95	0.769	0.737
SocraticLM	11.71	4.08	17.75	0.752	0.728	11.98	4.26	17.98	0.774	0.737	11.74	4.15	17.87	0.783	0.756
LEAP	**13.06**	**5.54**	**19.92**	**0.847**	**0.781**	**13.35**	**5.76**	**20.07**	**0.862**	**0.796**	**13.64**	**5.95**	**20.21**	**0.872**	**0.814**

Table 2: Experimental results of the automatic evaluation for guidance generation on LEAD dataset.

Methods	DS / t↓	CPP / t ↓	SA / t ↓
GPT-4o	2.74	2.82	2.45
Claude-3.5	2.68	2.76	2.35
EduChat	2.82	2.90	2.65
SocraticLM	2.92	3.18	2.80
LEAP	**2.36**	**2.50**	**1.95**

Table 3: Comparison of teaching effectiveness. 't' represents average turns used in each dataset.

Methods	B-2	B-3	R-L	Coh.	Ins.
LEAP	**14.75**	**6.36**	**21.11**	**0.782**	**0.826**
-Cog. Ext.	13.72	5.36	19.36	0.708	0.748
-Scf. Pln.	14.03	5.56	19.78	0.724	0.769
-Direct.	14.47	5.89	20.25	0.746	0.785
-MCTS.	14.28	5.72	19.96	0.734	0.773

Table 4: Ablation study on data structure dataset.

Comparison on Dialogue Quality

To demonstrate the effectiveness of our proposed LEAP, we first conduct the automatic evaluation of the generated guidance on three courses of LEAD dataset. Performance comparison of LEAP and baselines is shown in Table 2.

The experimental results demonstrate that our approach achieves a notable improvement over the baselines. Specifically, in Session 3 and Session 4, our method outperforms the baseline methods, highlighting its effectiveness in recording student's long-term learning process and planning scaffolds. Additionally, EduChat and SocraticLM exhibit poor generalizability; in contrast, our method demonstrates stronger generalization, enabling personalized teaching for different students across diverse educational scenarios.

Comparison on Teaching Effectiveness

To validate the tutoring dialogue system's ability to guide student thinking and ultimately lead them to solve the problem, we conduct simulation experiments. We build an agent-based interactive system to simulate interactions between a teacher and a student. In this setup, our proposed method serves as the teacher, while the student is simulated by the LLM. We use the average number of dialogue turns as the evaluation metric. A correct answer given earlier indicates better guidance ability of the model, so fewer turns represent better performance. We compared GPT-4o, Claude-3.5-sonnet (denoted as Claude-3.5), EduChat, SocraticLM, and LEAP on tutoring tasks across three courses.

Table 3 shows the results of teaching effectiveness. LEAP achieved the best performance, validating its effectiveness in real-world scenarios. It demonstrates the ability to guide students to understand key concepts more quickly and answer questions correctly.

Ablation Study

To validate the effectiveness of different components of our proposed framework, we conduct ablation study to validate the components used in LEAP. Specifically, we remove the cognitive state extraction module and the scaffolding instruction planning module, respectively. When the scaffolding instruction planning is removed, we directly use the cognitive state and context for generation. For the scaffolding instruction planning module, we further remove the direct planner and MCTS planner to validate the effectiveness. It means that we use the alternative planner for planning in all cases, directly generating responses based on the cognitive

Proportion	Prec.	B-3	R-L	Call.↓
0.0%	0.682	5.72	19.96	**1,526**
27.5%	0.766	5.83	20.17	2,847
52.7%	**0.872**	**6.36**	**21.11**	4,732
76.4%	0.814	5.95	20.38	6,124
100.0%	0.837	5.89	20.25	7,892

Table 5: Comparison of different MCTS proportions on data structure dataset. 'Prec.' denotes precision and 'Call.' denotes LLM calls.

state and context.

Table 4 demonstrates the results of the ablation study on data structure dataset in session 4, comparing the model performance after removing specific components (denoted as '-'). For example, '-Cog. Ext.' denotes removing cognitive state extraction module. The results demonstrate that the performance of LEAP declines when removing these components, validating the significance of these modules.

Performance of Different MCTS Proportions

To validate the performance of different MCTS proportions, we conduct experiments on data structure dataset in session 4. The proportion listed we chose represents the actual MCTS applied ratio under settings of 0%/25%/50%/75%/100%.

Table 5 presents the performance of LEAP under different MCTS proportions, including the precision of scaffold planning and the quality of the generated dialogues. In addition, we compare the number of LLM calls under different proportions. Since the primary inference cost lies in the number of LLM calls, this metric also reflects the efficiency of the method. The experimental results show that the selected proportion achieves the best performance. Compared to higher proportions, our approach achieves both cost reduction and performance improvement. Although a higher MCTS proportion allows the model to engage in more reasoning, it leads to a decline in performance during the scaffolding instruction planning process. This suggests that excessive reasoning, when a scaffold already aligns well with the cognitive state, can be detrimental, consistent with findings from existing research (Ma et al. 2023).

Performance of Generalizability

Additionally, we conducted cross-course experiments on data structure and C++ programming datasets in session 4 to validate generalizability of our proposed model. We use models trained on another dataset to test on the current dataset. In this setup, we use 'DS-LEAP' to represent the model trained on DS and use 'CPP-LEAP' to represent the model trained on CPP. We also compared them with ICL prompting (denoted as prompting)

As shown in Figure 4, with the cross-course setup, LEAP demonstrated strong generalization ability: the model trained on the CPP dataset also performed well on the DS course, and vice versa. This indicates that our approach performs well across different education scenarios.

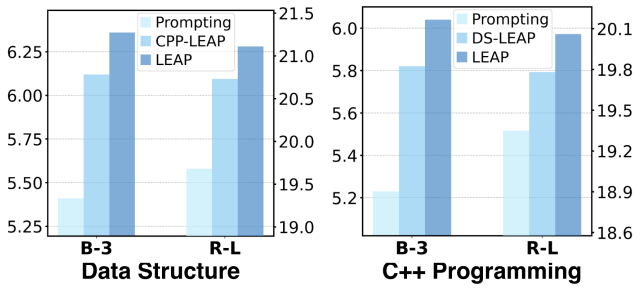

Figure 4: Performance of cross-course experiments on data structure and C++ programming dataset.

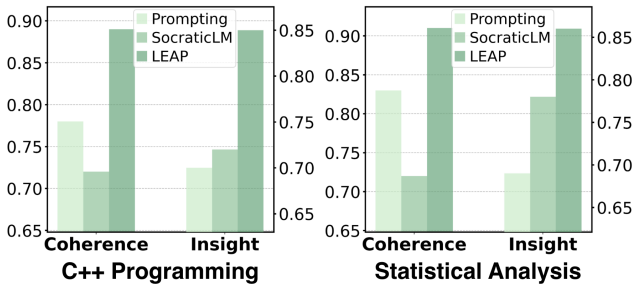

Figure 5: The results of human evaluation on C++ programming and statistical analysis dataset.

Human Evaluation

To assess the quality of the generated guidance from a human perspective, we also conducted human evaluation. We recruited five Ph.D. students in the field of computer science to participate in the evaluation. They were asked to rate the selected tutoring dialogues based on the given dialogue context along coherence and insight, using a scoring scale from 0 to 1. We randomly selected 100 generated dialogue rounds in C++ programming and statistical analysis dataset and utilize ICL prompting (denoted as prompting) and SocraticLM as baselines. We use the average of these scores as the final result.

As shown in Figure 5, our approach outperforms the baseline methods in both coherence and insight, validating its effectiveness. Prompting yields coherent but lacks insight compared to SocraticLM and our LEAP, underscoring LEAP's superior adaptive guidance. Prompting and SocraticLM both fell short in insight, whereas LEAP's robust performance demonstrates its effectiveness across difficulty levels.

Conclusions

In this work, we propose LEAP, a long-term tutoring dialogue system to address the lack of attention on the long-term learning process of students on tutoring dialogue systems. It captures students' long-term learning performance through cognitive states extraction module and predict scaffolding instructions through scaffolding instruction planning module. We introduce LEAD, a long-term tutoring dialogue dataset based on real-world student multi-round submissions.

Acknowledgments

This research was partially supported by grants from the National Natural Science Foundation of China (No.62037001, No.62307032), and the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant No. 2025C02022.

References

Abdelghani, R.; Wang, Y.-H.; Yuan, X.; Wang, T.; Lucas, P.; Sauzéon, H.; and Oudeyer, P.-Y. 2024. GPT-3-driven pedagogical agents to train children's curious question-asking skills. *International Journal of Artificial Intelligence in Education*, 34(2): 483–518.

Caines, A.; Yannakoudakis, H.; Edmondson, H.; Allen, H.; Pérez-Paredes, P.; Byrne, B.; and Buttery, P. 2020. The Teacher-Student Chatroom Corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, 10–20.

Chen, J.; Liu, Z.; Hou, M.; Zhao, X.; and Luo, W. 2024. Multi-turn Classroom Dialogue Dataset: Assessing Student Performance from One-on-one Conversations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, 5333–5337. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704369.

Dai, Z.; Chen, B.; Zhao, Z.; Tang, X.; Wu, S.; Yao, C.; Gao, Z.; and Chen, J. 2025. Less is More: Adaptive Program Repair with Bug Localization and Preference Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 128–136.

Dai, Z.; Yao, C.; Han, W.; Yuanying, Y.; Gao, Z.; and Chen, J. 2024. Mpcoder: Multi-user personalized code generator with explicit and implicit style representation learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3765–3780.

Dan, Y.; Lei, Z.; Gu, Y.; Li, Y.; Yin, J.; Lin, J.; Ye, L.; Tie, Z.; Zhou, Y.; Wang, Y.; et al. 2023. Educhat: A large-scale language model-based chatbot system for intelligent education. *arXiv preprint arXiv:2308.02773*.

Deng, Y.; Zhang, W.; Lam, W.; Ng, S.-K.; and Chua, T.-S. 2024. Plug-and-Play Policy Planner for Large Language Model Powered Dialogue Agents. In *ICLR*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Gao, W.; Liu, Q.; Yue, L.; Yao, F.; Lv, R.; Zhang, Z.; Wang, H.; and Huang, Z. 2025. Agent4Edu: Generating Learner Response Data by Generative Agents for Intelligent Education Systems. *arXiv preprint arXiv:2501.10332*.

He, T.; Liao, L.; Cao, Y.; Liu, Y.; Liu, M.; Chen, Z.; and Qin, B. 2024. Planning Like Human: A Dual-process Framework for Dialogue Planning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4768–4791. Bangkok, Thailand: Association for Computational Linguistics.

Hirunyasiri, D.; Thomas, D. R.; Lin, J.; Koedinger, K. R.; and Aleven, V. 2023. Comparative analysis of gpt-4 and human graders in evaluating praise given to students in synthetic dialogues. *arXiv preprint arXiv:2307.02018*.

Howe, C.; Hennessy, S.; Mercer, N.; Vrikki, M.; and Wheatley, L. 2019. Teacher–student dialogue during classroom teaching: Does it really impact on student outcomes? *Journal of the learning sciences*, 28(4-5): 462–512.

Hu, L.; Dong, Z.; Chen, J.; Wang, G.; Wang, Z.; Zhao, Z.; and Wu, F. 2023. PTADisc: a cross-course dataset supporting personalized learning in cold-start scenarios. *Advances in Neural Information Processing Systems*, 36: 44976–44996.

Huang, H.; Niu, T.; Yang, R.; and Shi, L. 2025. RAM2C: A Liberal Arts Educational Chatbot based on Retrieval-augmented Multi-role Multi-expert Collaboration. In Rambow, O.; Wanner, L.; Apidianaki, M.; Al-Khalifa, H.; Eugenio, B. D.; and Schockaert, S., eds., *Proceedings of the 31st International Conference on Computational Linguistics*, 448–458. Abu Dhabi, UAE: Association for Computational Linguistics.

Kong, C.; Fan, Y.; Wan, X.; Jiang, F.; and Wang, B. 2024. PlatoLM: Teaching LLMs in Multi-Round Dialogue via a User Simulator. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7841–7863. Bangkok, Thailand: Association for Computational Linguistics.

Kwon, S.; Kim, S.; Park, M.; Lee, S.; and Kim, K. 2024. BIPED: Pedagogically Informed Tutoring System for ESL Education. *arXiv preprint arXiv:2406.03486*.

Li, H.; Yang, C.; Zhang, A.; Deng, Y.; Wang, X.; and Chua, T.-S. 2024. Hello Again! LLM-powered Personalized Agent for Long-term Dialogue. *arXiv preprint arXiv:2406.05925*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, J.; Huang, Z.; Xiao, T.; Sha, J.; Wu, J.; Liu, Q.; Wang, S.; and Chen, E. 2024a. SocraticLM: Exploring Socratic Personalized Teaching with Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liu, Z.; Yin, S. X.; Lin, G.; and Chen, N. F. 2024b. Personality-aware Student Simulation for Conversational Intelligent Tutoring Systems. *arXiv preprint arXiv:2404.06762*.

Long, L.; Wang, R.; Xiao, R.; Zhao, J.; Ding, X.; Chen, G.; and Wang, H. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Lv, X.; Li, M.; Chen, J.; Dong, Z.; Han, S.; and Liao, B. 2025a. Out-of-Distribution Detection via LLM-Guided Outlier Generation for Text-attributed Graph. In *Findings of the Association for Computational Linguistics: ACL 2025*, 19544–19555.

Lv, X.; Wang, G.; Chen, J.; Su, H.; Dong, Z.; Zhu, Y.; Liao, B.; and Wu, F. 2025b. Debiased Cognition Representation

Learning for Knowledge Tracing. *ACM Transactions on Information Systems*.

Ma, Y.; Cao, Y.; Hong, Y.; and Sun, A. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 10572–10601. Singapore: Association for Computational Linguistics.

Macina, J.; Daheim, N.; Chowdhury, S.; Sinha, T.; Kapur, M.; Gurevych, I.; and Sachan, M. 2023a. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 5602–5621. Singapore: Association for Computational Linguistics.

Macina, J.; Daheim, N.; Wang, L.; Sinha, T.; Kapur, M.; Gurevych, I.; and Sachan, M. 2023b. Opportunities and Challenges in Neural Dialog Tutoring. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2357–2372. Dubrovnik, Croatia: Association for Computational Linguistics.

Nye, B. D.; Graesser, A. C.; and Hu, X. 2014. AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24: 427–469.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Park, M.; Kim, S.; Lee, S.; Kwon, S.; and Kim, K. 2024. Empowering Personalized Learning through a Conversation-based Tutoring System with Student Modeling. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24. New York, NY, USA: Association for Computing Machinery. ISBN 9798400703317.

Phung, T.; Pădurean, V.-A.; Cambronero, J.; Gulwani, S.; Kohn, T.; Majumdar, R.; Singla, A.; and Soares, G. 2023. Generative AI for programming education: benchmarking ChatGPT, GPT-4, and human tutors. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*, 41–42.

Qiu, Y.; Deng, Y.; Yao, Q.; Zhang, Z.; Dong, Z.; Yao, C.; and Chen, J. 2025. Think both ways: Teacher-student bidirectional reasoning enhances MCQ generation and distractor quality. In *Findings of the Association for Computational Linguistics: ACL 2025*, 8240–8253.

Robertson, A. D.; Atkins, L. J.; Levin, D. M.; and Richards, J. 2015. What is responsive teaching? In *Responsive teaching in science and mathematics*, 1–35. Routledge.

Stasaski, K.; Kao, K.; and Hearst, M. A. 2020. CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 52–64.

Väth, D.; Vanderlyn, L.; and Vu, N. T. 2023. Conversational Tree Search: A New Hybrid Dialog Task. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 1264–1280. Dubrovnik, Croatia: Association for Computational Linguistics.

Wang, S.; Zhou, K.; Lai, K.; and Shen, J. 2020. Task-Completion Dialogue Policy Learning via Monte Carlo Tree Search with Dueling Network. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3461–3471. Online: Association for Computational Linguistics.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wiske, M. S. 1998. *Teaching for Understanding. Linking Research with Practice. The Jossey-Bass Education Series.* ERIC.

Wu, T.; Chen, J.; Lin, W.; Li, M.; Zhu, Y.; Li, A.; Kuang, K.; and Wu, F. 2025. Embracing Imperfection: Simulating Students with Diverse Cognitive Levels Using LLM-based Agents. *arXiv preprint arXiv:2505.19997*.

Xu, J.; Szlam, A.; and Weston, J. 2022. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5180–5197. Dublin, Ireland: Association for Computational Linguistics.

Xu, S.; Zhang, X.; and Qin, L. 2024. EduAgent: Generative Student Agents in Learning. *arXiv preprint arXiv:2404.07963*.

Yu, X.; Chen, M.; and Yu, Z. 2023. Prompt-Based Monte-Carlo Tree Search for Goal-oriented Dialogue Policy Planning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7101–7125. Singapore: Association for Computational Linguistics.

Yue, M.; Mifdal, W.; Zhang, Y.; Suh, J.; and Yao, Z. 2024. MathVC: An LLM-Simulated Multi-Character Virtual Classroom for Mathematics Education. *arXiv preprint arXiv:2404.06711*.

Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; and Chua, T.-S. 2024a. On Generative Agents in Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, 1807–1817. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704314.

Zhang, Z.; Zhang-Li, D.; Yu, J.; Gong, L.; Zhou, J.; Liu, Z.; Hou, L.; and Li, J. 2024b. Simulating classroom education with llm-empowered agents. *arXiv preprint arXiv:2406.19226*.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 19724–19731.