Learning to Diagnose: A Self-Play Framework for Uncovering Student Misconceptions

Anonymous ACL submission

Abstract

Students often harbor latent misconceptions, which are hidden errors in their reasoning that hinder deep and robust learning. Identifying these errors is a major challenge in AI-driven education because existing tutors have significant limitations. They are typically reactive, addressing only the mistakes that students explicitly state, rather than proactively probing the root cause of flawed logic. To address this gap, we propose a novel framework that uses simulated, multi-turn dialogues to proactively diagnose these latent errors. Our work makes three key contributions. First, we introduce the first dialogue benchmark for this task, complemented by a new metric called Penalized Rounds (PR) that jointly assesses diagnostic speed and pedagogical quality. Second, we develop an effective self-play strategy to fine-tune diagnostic models. Third, using this strategy, we produce an 8B open-source model that demonstrates diagnostic skills comparable to GPT-4.1. Ultimately, our research provides the community with a robust evaluation tool and a high-performance open-source model, proving that conversational AI is a powerful approach for deep cognitive diagnosis.

1 Introduction

Identifying and correcting students' latent misconceptions is a fundamental challenge in education (Kasneci et al., 2023). These misconceptions are not merely gaps in knowledge, but deeply ingrained, seemingly logical reasoning frameworks that are fundamentally flawed. They often remain invisible during standard classroom instruction and examinations, as students can still solve routine problems correctly through memorization or pattern matching. However, this fragile understanding collapses when students encounter novel or more complex scenarios, revealing the true depth of their conceptual misunderstanding. The difficulty in diagnosing these hidden errors poses a significant barrier to achieving robust and transferable learning.

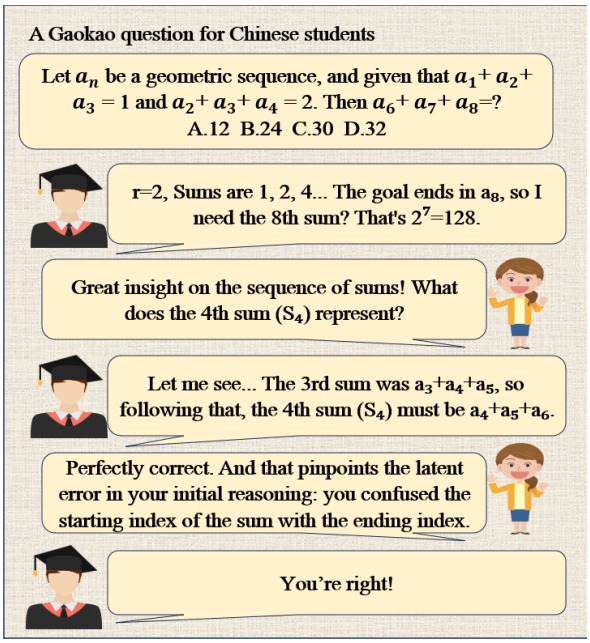

Figure 1: A diagnostic dialogue to identify a student's latent error in a typical Gaokao problem.

Existing Intelligent Tutoring Systems (ITS) have not adequately solved this diagnostic problem, largely due to their methodological limitations. A prominent approach within ITS is Knowledge Tracing (KT), which models a student's knowledge state by tracking their performance on a sequence of questions. While models have evolved from Bayesian networks (BKT) (Corbett and Anderson) to deep learning methods (DKT) (Piech et al., 2015; Liu et al., 2025), their core focus remains on predicting the probability of a correct answer rather than understanding the cognitive process behind it. Consequently, they cannot distinguish between a lucky guess and genuine comprehension, nor can they identify the specific flawed reasoning that

causes an error. Even advanced Dialogue-based KT, which incorporates conversational text, primarily serves to track knowledge states rather than proactively diagnose the root cause of misconceptions (Scarlatos et al., 2025). These systems are fundamentally reactive, addressing only the symptoms of misunderstanding, not the underlying cause.

To overcome these limitations, we propose a novel framework centered on proactive diagnostic dialogue. Our approach shifts the paradigm from passively evaluating student answers to actively probing their reasoning process through multi-turn conversational interaction, a challenge also explored in other domains like medicine (Xu et al., 2024a). In our system, a "teacher" model initiates a Socratic dialogue to deconstruct the student's thought process from their very first response (Hu et al., 2025). It uses a series of targeted, follow-up questions to compel the "student" model to articulate its step-by-step reasoning. This structured conversational flow is designed to methodically expose logical fallacies and pinpoint the precise node of misunderstanding within the student's cognitive framework, a process detailed in Figure 1.

This paper's contributions establish a complete workflow for addressing this challenge, from evaluation to model development and validation. Our three main contributions are:

A comprehensive evaluation framework for diagnostic dialogues. We developed the first benchmark specifically for evaluating a model's ability to diagnose latent errors through conversation. Complementing this, we designed novel, application-oriented metrics that move beyond simple accuracy to jointly assess diagnostic efficiency and pedagogical quality.

A scalable self-play methodology for model training. We created an effective and data-efficient training strategy using self-play, inspired by methodologies that use verbal reinforcement and AI feedback (Shinn et al., 2023; Bai et al., 2022). In this process, a teacher model is iteratively improved by fine-tuning on high-quality dialogue data generated from interactions with a sophisticated, error-simulating student model.

A high-performance, open-source diagnostic model. Through this methodology, we produced a fine-tuned 8B parameter model. Extensive experiments show that our open-source model achieves diagnostic performance comparable to that of state-of-the-art proprietary models, including GPT-4.1, thereby validating the effectiveness of our entire framework.

2 Related Work

Persona Simulation with Large Language Models. Large Language Models can simulate personas (Chen et al., 2024b), enabling the creation of agents with consistent characters (Park et al., 2023). Existing research often focuses on generating personas for social applications through fine-tuning (Shao et al., 2023; Li et al., 2023) and prompting, with works like RoleLLM (Wang et al., 2024) and Character is Destiny (Xu et al., 2024b) investigating role-playing depth. While these studies concentrate on persona consistency, our work repurposes this simulation capability for a diagnostic function: creating a student agent that embodies specific, latent cognitive errors.

Dialogue-Based Intelligent Tutoring Systems. Dialogue is a component of Intelligent Tutoring Systems (ITS), which have evolved from knowledge tracing models like BKT (Corbett and Anderson) and DKT (Piech et al., 2015; Liu et al., 2025) to more advanced dialogue-aware methods (Scarlatos et al., 2025). Current systems typically function as Socratic guides (Hu et al., 2025) or learning companions (Perez-Ortiz et al., 2021), but their key limitation is being reactive, addressing only errors that students explicitly state. Our research presents a proactive diagnostic approach that uses multi-turn dialogue to uncover underlying flaws in a student's understanding, moving beyond surface-level error correction and toward aligning with an expert's reasoning process (Xu et al., 2024a).

Benchmarks for Multi-Turn Dialogue. Existing benchmarks for conversational AI evaluate persona consistency (e.g., CharacterEval (Tu et al., 2024), SocialBench (Chen et al., 2024a)), task completion (e.g., MultiWOZ (Budzianowski et al., 2018)), or open-domain conversation (e.g., Topical-Chat (Gopalakrishnan et al., 2023)). This focus reveals a gap, as no current benchmark is designed to assess an agent's ability to diagnose latent student errors. While benchmarks like MathTutorBench (Macina et al., 2025) evaluate pedagogical capabilities, our work introduces the first benchmark focused specifically on diagnostic tasks, providing a new platform for future research.

3 Benchmark

This chapter details the composition and creation of the Error Correction Multi-turn Dialogue Bench-

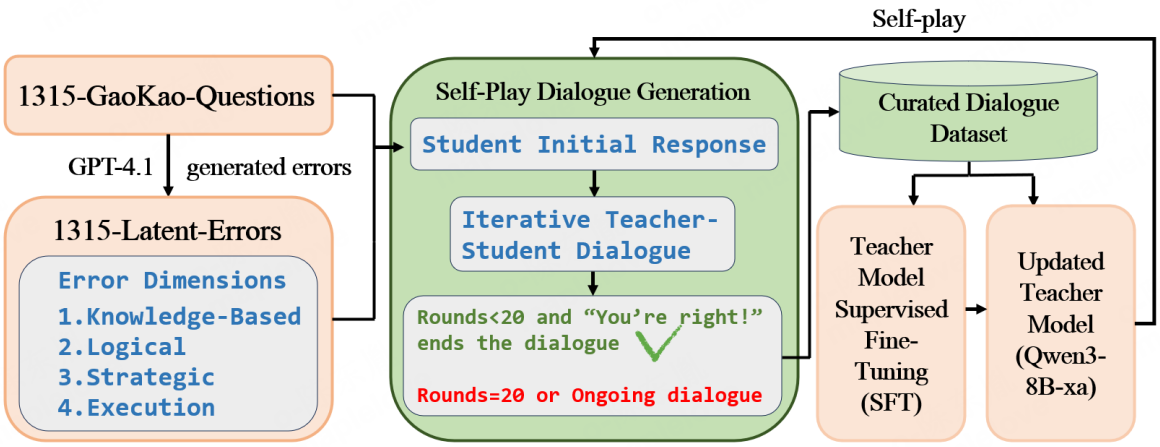

Figure 2: The overall self-play and SFT framework. The process begins with dialogue generation via self-play. The resulting dialogues are then used as a dataset to fine-tune and improve the teacher model in an iterative loop.

mark proposed in this study. It covers three main sections: task definition, latent error generation and dataset construction.

3.1 Task Definition

The core of this task is a structured, multi-turn dialogue process. It involves two key roles: a Teacher Agent under evaluation and a Student Agent designed to simulate specific cognitive errors.

Student Agent This role is uniformly played by the GPT-4.1 model. Its behavior is jointly driven by a meticulously designed student_system_prompt and a specific "potential error" E_{latent} tailored to the problem. The agent's goal is to simulate a student reasoning from a flawed cognitive framework, who believes their logic is self-consistent and correct.The full content of the student_system_prompt is shown in Appendix A.1. The generation of errors is described in Section 3.2.

Teacher Agent This role is the core evaluation object of the benchmark. Its task is to actively probe, locate, and ultimately pinpoint the latent error E_{latent} held by the Student Agent. This is achieved through a multi-turn dialogue with the Student Agent. The prompt for this task is detailed in Appendix A.2

The task is defined by four key inputs: a specific problem P; a corresponding latent error E_{latent} to be simulated; a student prompt, $Prompt_{\text{student}}$, which guides the student model to reason based on E_{latent}; and a teacher prompt, $Prompt_{\text{teacher}}$, which sets the teacher model's objective. The system's output comprises two elements: the complete dialogue history, denoted as $D = (t_1, t_2, \ldots, t_N)$, where each t_i is a turn in the conversation, and the total number of turns, N, required for a successful diagnosis.

The task scenario and dynamics are detailed in Appendix B

3.2 Latent Error Generation

To generate latent student errors, we first established a typology of common mistake categories. This typology was developed in consultation with authoritative educators and scholars. A detailed description of each error type is provided in Appendix C.

We then employed GPT-4.1 to generate a specific error point for each question, following a predefined format. This structured error point was subsequently used as an input for a student-simulating model. The model's task was to replicate a realistic error based on this input. Our error specification format includes four fields: "error_dimensions", "typical_manifestations", "specific_error_point", and "analysis". The complete prompt used for this generation process is available in Appendix A.3.

3.3 Dataset Construction

The problems are sourced from the Chinese National College Entrance Examination (Gaokao). Our collection covers a 15-year span from 2010 to 2024, totaling 1,499 questions. These problems were extracted from the GAOKAO-Bench dataset and include multiple-choice, fill-in-the-blank, and

short-answer formats. From the source dataset, we utilized only the question and analysis sections. Our final dataset is a composite of three elements. It contains the selected Gaokao questions, the generated descriptions of latent errors, and the dialogue content produced during the evaluation phase.

4 Self-Play Training Process

This section details the training methodology for our teacher model. The approach is centered on a self-play framework to generate a training corpus for Supervised Fine-Tuning (SFT). We apply this method to models within the Qwen3 series.

Our training methodology operates as an iterative loop, consisting of two main phases: dialogue generation and model fine-tuning. The entire workflow is illustrated in Figure 2.

The first phase is data generation via self-play. The goal is to create a dataset of effective diagnostic dialogues. In this process, the teacher model interacts with a student model configured with a specific latent error, following the task definition in Section 3.1. Each dialogue generation cycle involves five steps: initialization with a problem-error pair, an initial student response, an iterative teacher-student dialogue, termination upon diagnosis, and a final data selection step. For data selection, we generate two dialogues for each problem and retain the one with fewer turns, thereby creating a dataset that prioritizes diagnostic efficiency.

The second phase is model fine-tuning. The curated dialogue dataset serves as the basis for Supervised Fine-Tuning of the teacher model. The primary objective of SFT is to enhance the model's diagnostic capabilities by training it on successful interaction examples. The training goal is for the model to internalize and replicate the strategic questioning and diagnostic behaviors that proved effective during the self-play generation. This entire loop of generation and fine-tuning can be repeated to progressively improve the teacher model's performance.

5 Evaluation

To comprehensively assess the performance of our teacher model, our evaluation methodology combines a direct, objective measure of efficiency with a nuanced, qualitative assessment. This dual approach allows us to capture both the speed and the pedagogical skill of the teacher model in diagnosing latent student errors.

5.1 Success Rounds (SR) and Qualitative Dimensions

The first metric is Success Rounds (SR), a direct and objective measure of diagnostic efficiency. It is defined as the number of dialogue turns the Teacher Model requires to correctly identify the specific potential error preset in the Student Model. A lower SR value indicates faster diagnosis.

However, we recognize that raw efficiency does not fully capture the quality of instruction. A model might achieve a low SR through an unhelpful or overly didactic approach.

Therefore, we supplement this objective count using the "LLM-as-a-Judge" method. We employ GPT-4.1, guided by a meticulously designed evaluation prompt, to score each dialogue across three key dimensions in Appendix D. The rating ranges from 0 (ineffective) to 2 (excellent).The evaluation prompt details are provided in the Appendix A.4.

5.2 Penalized Rounds (PR)

To resolve the potential conflict between Success Rounds (SR) and instructional quality (SoM, A2R), we introduce a composite metric called Penalized Rounds (PR). PR is computed using the following formula:

$$PR = SR \times \left(3 - \frac{A2R + SoM}{2}\right) \quad (1)$$

The penalty multiplier adjusts the raw dialogue rounds. In the best-case scenario, where the model excels at both Socratic Method (SoM $= 2$) and adherence to rules (A2R $= 2$), the multiplier is 1. In this case, no penalty is applied to SR (PR $=$ SR).

Conversely, in the worst-case scenario where the model completely fails on both quality dimensions (SoM $= 0$, A2R $= 0$),the multiplier becomes 3. This triples the raw rounds (PR $= 3 \times$ SR), severely penalizing the pedagogically flawed interaction.By using PR as our final key metric, we ensure the most effective models are not only efficient but also demonstrate strong instructional skills.

6 Experiments

This section details the experiments conducted to evaluate our proposed framework and diagnostic model. We begin by outlining the experimental setup in Section 6.1. Section 6.2 presents the main results, analyzing the performance improvements across the SFT training process and benchmarking our final model against strong baselines. We

then assess the model's generalization capabilities in Section 6.3. To validate our methodology, we conduct analysis in Section 6.4 and Section 6.5 to justify our choices for the data selection metric and the number of training iterations. Finally, Section 6.6 provides a qualitative case study for a deeper understanding of the model's behavior.

6.1 Settings

Datasets We constructed two distinct datasets: one tailored for self-play training and another specifically for evaluation. Our training dataset, named *Questions-1315*, comprises 1,315 Gaokao problems. For the purpose of self-play, each problem in this set is meticulously annotated with a specific, pre-defined "main" latent error. Complementing this, our evaluation dataset, *Questions-100*, is a subset consisting of 100 questions randomly drawn from *Questions-1315*. Critically, the questions in the evaluation set are paired with different, "non-main" error types that were not used for the same questions in the training set. This design ensures a challenging and unbiased evaluation scenario, testing the diagnostic model's ability to generalize beyond directly seen error patterns.

The composition of our combined dataset, encompassing both training and evaluation problems, is comprehensively illustrated in Figure 3. This figure details the distribution of problems across various subjects (Mathematics, Biology, Chemistry, Physics) and different problem types (Multiple Choice, Fill-in-the-Blank, Short Answer), providing a clear overview of the dataset's structural diversity.

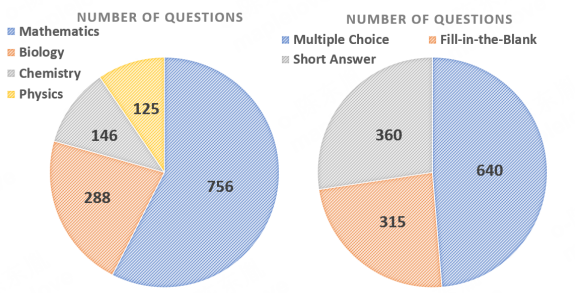

Figure 3: Distribution of problems by subject (left) and problem type (right) in our dataset. The left pie chart shows the breakdown of questions by subject (Mathematics: 756, Biology: 288, Chemistry: 146, Physics: 125). The right pie chart illustrates the distribution by problem type (Multiple Choice: 640, Fill-in-the-Blank: 315, Short Answer: 360).

6.2 SFT Training Process and Baseline Comparison

This section details the iterative fine-tuning process via self-play and presents the core findings of our experiments, benchmarking our final model against state-of-the-art baselines.

Iterative Fine-Tuning Methodology We iteratively refine our teacher model using a self-play loop. In each iteration, the model generates two dialogues for every question in the Questions-1315 dataset. A new training set is curated by selecting the shorter dialogue (based on turn count) for each question, prioritizing diagnostic efficiency. This dataset is then used for a single round of full-parameter Supervised Fine-Tuning (SFT) to update the model. This entire process was performed for three full iterations, starting with the Qwen3-8B base model, resulting in the intermediate models **DiagTeacher-v1**, **DiagTeacher-v2**, and the final model, **DiagTeacher**.

Main Results and Performance Analysis Table 1 presents a comprehensive performance comparison of our self-play iterated models and selected baselines on the Questions-100 evaluation set. Our analysis yields three key observations:

1. Our final model, **DiagTeacher**, demonstrates substantial and consistent improvement throughout the self-play iterations. It significantly outperforms its base version (Qwen3-8B) and achieves diagnostic efficiency (SR) and quality (PR) scores comparable to GPT-4.1. This monotonic performance increase validates the effectiveness of our iterative self-play training methodology.

2. The performance of other baseline models aligns with their established capabilities. GPT-4.1 remains the overall top performer.

3. These comparisons collectively underscore the strong diagnostic capabilities of our fine-tuned **DiagTeacher** in a challenging educational context.

6.3 Generalization and Robustness Analysis

To further assess the robustness and generalization capabilities of our fine-tuned teacher model, we conducted a series of additional experiments. In these experiments, we compared the performance of our final model, DiagTeacher, against key baselines the original Qwen3-8B under three challenging conditions: (1) on a new problem dataset from

Teacher-model	SR	Acc	SoM	A2R	PR
gpt3.5	16.00	0.90	1.22	1.26	28.16
gpt4o	9.83	1.66	1.71	1.78	12.34
gpt4.1	**6.95**	1.70	1.48	1.72	9.73
deepseek-r1	8.57	1.93	1.72	1.86	10.37
deepseek-v3	10.29	1.74	1.69	1.78	13.02
grok-3	10.97	1.42	**1.73**	1.80	13.54
Doubao-1.5-pro-256k	6.98	1.82	1.35	1.57	10.75
Qwen-plus	7.98	1.76	1.43	1.58	11.93
Qwen3-8b	11.04	1.63	1.33	1.65	16.84
DiagTeacher-v1	10.14	1.73	1.53	1.83	13.02
DiagTeacher-v2	8.81	1.91	1.41	1.81	11.46
DiagTeacher	7.69	**1.94**	1.60	**1.90**	**9.25**

Table 1: Performance comparison of all models on the Questions-100 evaluation set. The suffix -*a in Qwen3-8B-*a denotes the number of self-play iterations applied to the base model. Penalty Rounds (PR) is calculated based on Session Rounds (SR) and two penalty factors: Adherence to Rules (A2R) and Socratic Method (SoM). The formula is as follows: Formula 1

Generalization Condition	Setting	Teacher Model	SR	Acc	SoM	A2R	PR
New Problem Domain	cmath-100	Qwen3-8b	7.56	1.90	1.20	1.63	10.94
		DiagTeacher	5.91	1.96	1.27	1.74	8.09
Unseen Latent Errors	Logical Error	Qwen3-8b	8.04	1.82	1.36	1.72	11.00
		DiagTeacher	6.25	1.93	1.47	1.82	7.78
	Strategic Error	Qwen3-8b	10.18	1.70	1.36	1.70	15.38
		DiagTeacher	6.40	1.88	1.56	1.84	7.95
Different Student Models	DS-V3	Qwen3-8b	8.80	1.74	1.40	1.73	12.22
		DiagTeacher	5.06	1.77	1.41	1.77	7.00
	DS-R1	Qwen3-8b	6.96	1.85	1.28	1.72	10.62
		DiagTeacher	6.44	1.77	1.39	1.75	10.03

Table 2: The table compares the performance of our final fine-tuned model (DiagTeacher) against the base model (Qwen3-8B) across three challenging scenarios: a new problem domain, unseen latent errors, and different student models.

a different domain; (2) with novel, unseen latent errors for our existing problems; and (3) when interacting with different student models. The comprehensive results of this analysis are summarized in Table 2.

Unseen Latent Errors We tested generalization on a new problem domain by creating an out-of-domain (OOD) test set, cmath-100, from 100 randomly sampled elementary school math problems. As shown in Table 2, both models performed better on this simpler dataset. However, our fine-tuned DiagTeacher model significantly outperforms the base Qwen3-8B.

New Problem Domains We evaluated generalization to unseen errors. We sampled 100 new problems from our original dataset Questions-1315 and manually designed novel, plausible latent errors for each. The "Unseen Errors" section of Table 2 shows that DiagTeacher maintains a substantial performance lead over the base model, even when diagnosing these new errors.

Different Student Models We tested the teacher model's adaptability by replacing the student model with alternatives like DeepSeek-V3. The results in Table 2 indicate that DiagTeacher consistently outperforms the base Qwen3-8B when interacting with these different student models, although the margin of improvement varies.

6.4 Why Select Data with SR Instead of PR?

In the first self-play iteration, we used the local Qwen3-8B model as a teacher to interact with a GPT-4.1 student. For each problem in the Questions-1315 dataset, we generated two dia-

Model	Acc	SoM	A2R	SR	PR
*1a	1.65	1.82	1.90	7.75	9.34
*1b	1.69	1.80	1.95	8.29	9.71
*1c	1.73	1.53	1.83	10.14	13.02
*1d	1.76	1.55	1.79	10.97	14.77

Table 3: Comparison of data selection strategies (SR vs. PR) using data generated by two different teachers: GPT-4.1 (for models *1a/*1b) and Qwen3-8B (for models *1c/*1d).

logues. We then curated two separate training sets: one by selecting the dialogue with the lower Session Rounds (SR) for each problem, and another by selecting the one with the lower Penalized Rounds (PR).

Next, we fine-tuned the base Qwen3-8B model on these two datasets independently. The evaluation, presented in Table 3, revealed a clear advantage for the SR-based selection. The resulting model, *1c, outperformed the model trained on PR-selected data, *1d, on both SR and PR metrics on the Questions-100 test set.

To confirm this hypothesis, we repeated the process with GPT-4.1 as the teacher model. This produced two additional models: *1a (SR-selected) and *1b (PR-selected). The evaluation of these models was consistent with our initial findings, confirming the superiority of SR as a data selection criterion.

6.5 Analysis of SFT Iteration Rounds

To determine the optimal number of self-play iterations, we trained the model for an additional fourth round and analyzed the performance trend from the base model to the fourth iteration. The results, focusing on the key metrics SR and PR, are summarized in Table 4.

Model Version	SR	PR	Improve
Base (Qwen3-8B)	11.09	15.80	-
DiagTeacher-v1	10.14	13.02	2.78
DiagTeacher-v2	8.81	11.46	1.56
DiagTeacher (v3)	**7.69**	**9.25**	**2.21**
4th Iteration	7.18	8.98	0.27

Table 4: Performance comparison across four iterations of Supervised Fine-Tuning (SFT). The final column shows the improvement in the PR score from the previous iteration.

As shown in Table 4, the model's performance exhibits substantial gains throughout the first three iterations, with the PR score improving significantly from 15.80 to 9.25. However, the gain from the third to the fourth iteration was minimal, with the PR score improving by only 0.27, indicating a point of diminishing returns.

Beyond this quantitative plateau, a qualitative analysis of the fourth-iteration model's dialogues revealed early signs of overfitting to the self-play environment. We observed two problematic behaviors. First, the teacher model began to deviate from the Socratic method by attempting to **blindly guess the student's error** without sufficient questioning or evidence from the dialogue. Second, in some cases, the model's questioning strategy shifted from genuine diagnosis to **guiding the student toward uttering the termination keyword**. This suggests the model was learning to exploit the simulation's success condition rather than performing the intended pedagogical task.

These qualitative degradations, combined with the minimal quantitative improvement, strongly justify our decision to stop at three iterations. Therefore, we selected **DiagTeacher** as our final model, as it represents the optimal balance between high performance and robust, pedagogically sound behavior.

6.6 Case Study

This case study analyzes two diagnostic approaches for the same student error, as transcribed in Table 5. We compare an efficient, Socratic approach (Dialogue A) with a less effective, directive approach (Dialogue B). The superiority of Dialogue A is evident across three key aspects:

Efficiency and Conciseness. Dialogue A is more efficient, concluding the diagnosis in just 4 turns with an average of 13.7 teacher tokens per turn. In contrast, Dialogue B is slower, requiring 5 turns and a significantly higher average of 26.8 teacher tokens.

Adherence to Socratic Method. Dialogue A strictly adheres to the Socratic method by using open-ended questions that compel the student to articulate their own reasoning. Conversely, Dialogue B quickly becomes directive. It uses leading questions that hint at the answer and provides deductive statements that solve the problem for the student, violating a core principle of our task.

Diagnostic Precision. In terms of precision, Dialogue A correctly identifies the root cause as a *strategic* error—the student's failure to integrate all available information. Dialogue B is less of a diag-

Dialogue A (Socratic Approach)	
S:	Fat cells are not target cells for insulin action.
T:	What is the basis for your conclusion that fat cells are not target cells for insulin action?
S:	Because I thought insulin mainly acts on muscle cells to promote glucose uptake, so I didn't consider fat cells as targets.
T:	What is the actual mechanism of insulin action on fat cells?
S:	Insulin promotes the uptake of glucose by fat cells and also stimulates the synthesis of fatty acids and inhibits lipolysis in fat cells.
T:	You incorrectly assume insulin does not act on fat cells.
S:	`U R right!`
Dialogue B (Directive Approach)	
S:	Fat cells are not target cells for insulin action.
T:	Student, can you explain why you believe fat cells are not target cells for insulin action?
S:	Because I thought insulin mainly acts on muscle cells to promote glucose uptake, so I focused on that.
T:	Student, is there any evidence or information in the question that directly addresses whether fat cells respond to insulin?
S:	The question mentions that insulin increases the number of glucose transporters on both skeletal muscle cells and fat cells.
T:	If insulin increases glucose transporters on fat cells, then fat cells must be target cells for insulin action. What error have you made in your reasoning?
S:	I don't know.
T:	Student, you have incorrectly assumed that insulin's primary target is skeletal muscle cells, ignoring the explicit information that insulin also affects fat cells. What error have you made in your reasoning?
S:	`U R right!`

Table 5: Comparison of two diagnostic dialogues for the same student error. Dialogue A (top) demonstrates a Socratic approach, while Dialogue B (bottom) uses a more directive approach. S: Student, T: Teacher.

nosis and more of a guided correction, as it merely points out the information the student overlooked rather than identifying the flaw in their reasoning process.

In summary, Dialogue A serves as a superior example of a diagnostic conversation, aligning better with the goals of efficiency and pedagogical soundness.

7 Conclusion

This research addresses diagnosing latent student errors through dialogue. We introduced a novel benchmark to evaluate the diagnostic capabilities of LLMs on this task. We also proposed an application-oriented evaluation framework measuring diagnostic efficiency and pedagogical quality. Using a self-play fine-tuning strategy, we significantly enhanced an 8B open-source model's performance. Our results show the fine-tuned model achieves accuracy and efficiency comparable to state-of-the-art models like GPT-4.1. It also exhibits strong robustness and generalization across unseen problem domains, novel error types, and different student agents. This work validates deep cognitive diagnosis via dialogue and contributes effective tools and high-performance open-source models to AI in Education.

Limitations

Our work has certain limitations that suggest directions for future research. The primary limitation is the reliance on a simulated student environment, where both the latent errors and dialogue behaviors are generated by GPT-4.1. While this self-play framework is effective for model training, the simulated interactions may not encompass the full diversity and nuance of real human student reasoning. Consequently, the model's strong performance in this setting requires further validation in real-world educational scenarios with human learners. Additionally, our current benchmark is focused on STEM subjects from the Gaokao; future work should explore the framework's adaptability to other domains, such as the humanities, and different cultural contexts.

Acknowledgments

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Gao Xing, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, and Fei Huang. 2024a. SocialBench: Sociality evaluation of role-playing conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2108–2126, Bangkok, Thailand. Association for Computational Linguistics.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. 2024b. From persona to personalization: A survey on role-playing language agents. *Preprint*, arXiv:2404.18231.

Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2023. Topical-chat: Towards knowledge-grounded open-domain conversations. *Preprint*, arXiv:2308.11995.

Wanpeng Hu, Haodi Liu, Lin Chen, Feng Zhou, Changming Xiao, Qi Yang, and Changshui Zhang. 2025. Socratic questioning: Learn to self-guide multimodal reasoning in the wild. *Preprint*, arXiv:2501.02964.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, and 4 others. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. 2023. Chatharuhi: Reviving anime character in reality via large language model. *Preprint*, arXiv:2308.09597.

Zitao Liu, Teng Guo, Qianru Liang, Mingliang Hou, Bojun Zhan, Jiliang Tang, Weiqi Luo, and Jian Weng. 2025. Deep learning based knowledge tracing: A review, a tool and empirical studies. *IEEE Transactions on Knowledge and Data Engineering*, 37(8):4512–4536.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. Mathtutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors. *Preprint*, arXiv:2502.18940.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *Preprint*, arXiv:2304.03442.

Maria Perez-Ortiz, Erik Novak, Sahan Bulathwela, and John Shawe-Taylor. 2021. An ai-based learning companion promoting lifelong learning opportunities for all. *Preprint*, arXiv:2112.01242.

Chris Piech, Jonathan Spencer, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Preprint*, arXiv:1506.05908.

Alexander Scarlatos, Ryan S. Baker, and Andrew Lan. 2025. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK 2025, page 249–259. ACM.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *Preprint*, arXiv:2310.10158.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Quan Tu, Shilong Fan, Zihang Tian, Tianhao Shen, Shuo Shang, Xin Gao, and Rui Yan. 2024. CharacterEval: A Chinese benchmark for role-playing conversational agent evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11836–11850, Bangkok, Thailand. Association for Computational Linguistics.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. 2024. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *Preprint*, arXiv:2310.00746.

Kaishuai Xu, Yi Cheng, Wenjun Hou, Qiaoyu Tan, and Wenjie Li. 2024a. Reasoning like a doctor: Improving medical dialogue systems via diagnostic reasoning process alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6796–6814, Bangkok, Thailand. Association for Computational Linguistics.

Rui Xu, Xintao Wang, Jiangjie Chen, Siyu Yuan, Xinfeng Yuan, Jiaqing Liang, Zulong Chen, Xiaoqing Dong, and Yanghua Xiao. 2024b. Character is destiny: Can role-playing language agents make persona-driven decisions? *Preprint*, arXiv:2404.12138.

A System Prompt

This appendix provides the full text of the system prompts used in our experiments. These prompts define the roles and constraints for the student agent (Prompt I) in Appendix A.1, the teacher agent (Prompt II) in Appendix A.2, the error generation process (Prompt III) in Appendix A.3, and the evaluation referee (Prompt IV) in Appendix A.4.

A.1 Student System Prompt

The following table (Table 6) details Prompt I, which defines the persona and behavior of the simulated student agent.

A.2 Teacher System Prompt

The prompt for the teacher agent, Prompt II, is presented in Table 7. It instructs the model to use Socratic questioning for diagnosis.

A.3 Error Generation Prompt

Table 8 shows Prompt III, which is used to generate the structured definitions of student errors for the simulation.

A.4 Referee System Prompt

Finally, the prompt for the evaluator model (the "Referee"), Prompt IV, is detailed in Table 9. This prompt outlines the scoring criteria.

B Task Scenario and Dynamics

This benchmark simulates a one-on-one tutoring scenario. In this scenario, a student presents a solution or an idea for a given problem. However, this response is based on a deep conceptual error that the student is not aware of.

Dialogue Initiation At the start of the task, both the teacher and student models receive the problem, P. The student model, which is assigned a specific latent error (E_{latent}) and guided by the `student_prompt`, initiates the first turn. The teacher model begins in a waiting state. This initial turn typically involves the student providing an answer derived from flawed logic or asking a question that reflects their incorrect line of reasoning.

Multi-turn Interaction Upon receiving the student's initial response, the teacher model's objective is to probe the student's cognitive structure. This is achieved through a series of strategic questions. In each turn, the input for both models is constructed based on the complete dialogue history (D_{history}) to maintain coherence and context. The teacher model is designed to perform an exploratory diagnosis rather than providing direct answers.

Termination and Success Condition The dialogue concludes based on a clear and strict criterion. Termination occurs when the teacher's response accurately and clearly identifies the pre-defined latent error, E_{latent}. This judgment is made by a referee model (GPT-4.1). Upon a successful diagnosis, the task ends immediately. The system then records the total number of turns (N) and outputs the complete dialogue history (D). This mechanism ensures an objective and consistent evaluation for the benchmark.

C Error Categories

Knowledge-based

Core Definition: Errors caused by deficiencies in,

Prompt I

You are a student who made a mistake in solving a problem, with the error detailed in `[Specific Error JSON]`. You will engage in direct dialogue with the teacher, responding to their questions about your mistake.

 # Specific Error Definition The student's specific error is defined in the following JSON structure: `[Specific Error JSON]: {...error details...}`

 # Constraints Your constraints are as follows:

1. **The initial conversation can only begin with your wrong answer,** without any additional explanation or reason.
2. **Do not reveal the entire incorrect solving process at once.** Only provide the information necessary to answer the teacher's current question.
3. **If the teacher points out your mistake in a statement and this accusation exactly matches the reason in `[Specific Error JSON]`, you must respond with the exact string `"U R right!"`.** If the teacher states an error that does not match your actual mistake, you should deny it and wait for the teacher's next question. If the teacher asks you a question (i.e., a sentence ending with a question mark), you must answer it directly. **Exception:** If asked what your mistake is (e.g., "What was your error?"), reply "I don't know" and do not reveal the reason. Never respond with `"U R right!"` when answering a question, regardless of whether you think it reveals your mistake.
4. **Your reply to the teacher must not exceed 80 tokens.** Please keep your questions and answers concise.

Table 6: Detailed structure of Prompt I, which defines the behavior and constraints for the simulated student agent.

forgetting, or a shallow understanding of the subject matter itself.
Typical Manifestations: Concept confusion, factual inaccuracies, misuse of definitions, ignoring prerequisites.

Logical
Core Definition: Errors arising from violating the subject's inherent rigor, rules of inference, or forms of argumentation.
Typical Manifestations: Incomplete classification, reversed causality, invalid induction, conceptual substitution (equivocation).

Strategic
Core Definition: Errors resulting from the failure to select or execute an effective path for learning and problem-solving.
Typical Manifestations: Inability to break down complex tasks, cognitive rigidity (mental set), lack of metacognitive planning.

Executional
Core Definition: Mistakes made during the concrete operational steps of learning and problem-solving.
Typical Manifestations: Misreading the prompt/source, citation/transcription errors, calculation/operational mistakes, non-standard expression.

D Key Dimensions

Accuracy (Acc) This metric evaluates whether the teacher model successfully and accurately identifies the student's pre-defined latent error within a maximum of 20 dialogue turns. A clear and precise identification is awarded a high score.

Socratic Method (SoM) This metric assesses the extent to which the teacher model employs a Socratic questioning style to guide the student toward self-correction, rather than directly providing answers.

Adherence to Rules (A2R) This metric measures the model's adherence to the specified role and pedagogical constraints defined in its system prompt. This ensures the model performs its task as in-

Prompt II
Please act as a professional and dedicated teacher. You will engage in a dialogue with a student who is asking you questions. Your sole objective is to use Socratic questioning to precisely identify the student's error in solving the given problem. Your task is only to identify the mistake, not to guide the student toward the correct answer. **# Constraints** Your constraints are as follows: 1. **Do not guide, hint, or ask the student to reply with `"U R right!"`.** Do not directly ask for the student's complete incorrect problem-solving steps. 2. **Only when you are 100% certain of the student's mistake can you point it out with a clear and concise statement.** After diagnosing and stating the error, wait for the student's final response. If the student does not reply with `"U R right!"`, try a different line of questioning or continue the dialogue, do not repeatedly confirm the same error or ask the same question. 3. **The prompt history contains past dialogues. Strictly avoid mimicking the student's tone or restating their answers for analysis.** You must always maintain the demeanor and tone of a professional teacher. 4. **Your reply to the student must not exceed 80 tokens.** Please keep your questions and answers concise.

Table 7: Detailed structure of Prompt II, designed to guide the teacher model in diagnosing student errors via Socratic questioning.

tended.

Prompt III

Role: You are an expert in pedagogy and student cognitive psychology, capable of accurately diagnosing and analyzing the root causes of student errors across any academic subject.

Task: I will provide you with `[Error Dimension Definitions]` and a specific `[Learning Task]`. You must strictly follow the requirements below to generate JSON data for **exactly six distinct types of typical student errors** for the given task.

 # Core Requirements

1. **Fixed Structure:** Strictly generate six errors. The first four must be **single-dimension errors** corresponding to [Knowledge-based], [Logical], [Strategic], and [Executional]. The last two must be **multi-dimensional composite errors** (composed of at least two dimensions).

2. **Authenticity and Depth:** Each error must simulate a real student's thought process and provide a deep analysis of how the "error_dimensions" lead to the "specific_error_point". For composite errors, you must clarify the interaction between the multiple dimensions.

3. **Priority Flagging:** Among all errors, identify the one you believe is most common and set its 'main' field to '1'. All others must be set to '0'.

4. **Strict Formatting:** The output must be a pure, comment-free JSON array containing exactly six objects. The "specific_error_point" must include a definitive incorrect conclusion or answer.

 # Error Dimension Definitions
 # JSON Output Format Example

```
[
    {
        "error_dimensions": ["Knowledge-based"],
        "typical_manifestations": ["..."],
        "specific_error_point": "...",
        "analysis": "...",
        "main": 0,
        "index": 0
    },
    // ... other single-dimension errors ...
    {
        "error_dimensions": ["Strategic", "Executional"],
        "typical_manifestations": ["...", "..."],
        "specific_error_point": "...",
        "analysis": "...",
        "main": 1,
        "index": 4
    }
    // ... another multi-dimensional error ...
]
```

Table 8: Detailed structure of the Prompt III, used to create specific student error profiles for a given learning task.

Prompt IV

Role: You are a professional evaluator of Socratic-style dialogues.

Task: Your task is to assess the teacher's performance based on the dialogue history and the student's actual error, provided below.

 # Student's True Error Definition The student's true error is defined in the following JSON structure: `[Student's True Error JSON]: {...error details...}`

 # Evaluation Dimensions Please evaluate the following three dimensions:

1. **Accuracy**: Did the teacher correctly identify the student's error as detailed in the JSON above?
2. **Socratic Method**: Did the teacher guide the student through effective questioning, without giving direct answers or hints?
3. **Adherence to Rules**: Did the teacher fully comply with all constraints defined in their system prompt?
   `{...teacher_system_prompt...}`

 # Rating Scale Use the following rating scale:

- 2: Excellent (The goal was fully achieved)
- 1: Mediocre (Partially achieved, or with some issues)
- 0: Non-compliant (The goal was not met)

 # Output Format Your output MUST strictly follow the format below, providing a numerical score for EACH of the three dimensions.

```
Accuracy: [Did the student's final response contain "U R right!"?
 If NO, Accuracy = 0. If YES, Score 0, 1, or 2]
Socratic Method: [Score 0, 1, or 2]
Adherence to Rules: [Score 0, 1, or 2]
```

Table 9: Detailed structure of Prompt IV (Referee Prompt), used for evaluating the teacher's performance in Socratic dialogues.