BLCU-ICALL at BEA 2025 Shared Task: Multi-Strategy Evaluation of AI Tutors

Jiyuan An[1,2], Xiang Fu[1,2], Bo Liu[1,2], Xuquan Zong[1,2], Cunliang Kong[3], Shuliang Liu[4], Shuo Wang[3], Zhenghao Liu[4], Liner Yang[1,2,*], Hanghang Fan[1,2], Erhong Yang[5]

[1]National Language Resources Monitoring and Research Center for Print Media, Beijing Language and Culture University, China

[2]School of Information Science, Beijing Language and Culture University, China

[3]Department of Computer Science and Technology, Tsinghua University, China

[4]Department of Computer Science and Technology, Northeastern University, China

[5]Kika Tech, China

lineryang@gmail.com

Abstract

This paper describes our approaches for the BEA-2025 Shared Task on assessing pedagogical ability and attributing tutor identities in AI-powered tutoring systems. We explored three methodological paradigms: in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning from human feedback (RLHF). Results indicate clear methodological strengths: SFT is highly effective for structured classification tasks such as mistake identification and feedback actionability, while ICL with advanced prompting excels at open-ended tasks involving mistake localization and instructional guidance. Additionally, fine-tuned models demonstrated strong performance in identifying tutor authorship. Our findings highlight the importance of aligning methodological strategy and task structure, providing insights toward more effective evaluations of educational AI systems.

1 Introduction

The integration of large language models (LLMs) into educational technologies has revolutionized the landscape of AI-powered tutoring systems. These systems exhibit remarkable capabilities in generating fluent and contextually relevant responses, offering personalized learning experiences across various domains, including mathematics education. However, assessing the pedagogical effectiveness of these AI tutors extends beyond evaluating linguistic fluency or factual correctness; it necessitates a comprehensive analysis of their instructional strategies and their ability to engage students meaningfully.

To tackle the challenge of evaluating instructional quality, the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025) introduced a shared task titled Pedagogical Ability Assessment of AI-powered Tutors (Kochmar et al., 2025). This initiative aims to establish standardized evaluation criteria for systematically assessing the pedagogical effectiveness of AI-assisted educational dialogues. The task provides a unified evaluation framework encompassing four key pedagogical dimensions: mistake identification, mistake localization, provision of guidance, and actionability of feedback. In addition to these core dimensions, the shared task includes a fifth track, Guess the Tutor Identity, which focuses on authorship attribution by determining whether a response was generated by a specific language model or a human tutor—thereby shedding light on the stylistic signatures of different LLMs. An overview of the task design is illustrated in Figure 1.

In this paper, we present our comprehensive approach to the BEA-2025 Shared Task, focusing on both pedagogical ability assessment and tutor identity attribution in AI-powered tutoring systems. We explore multiple methodological paradigms, including in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RLHF), and demonstrate their respective strengths across task tracks. Our empirical results show that SFT excels in structured classification tasks, while ICL, supported by advanced prompting strategies, proves more effective in open-ended reasoning settings. Furthermore, we validate the use of fine-tuned LLM classifiers for authorship attribution, achieving competitive performance even in black-box conditions. Our findings not only highlight the importance of methodological alignment with task structure but also provide practical insights into building robust evaluation systems for educational AI.

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1084–1097

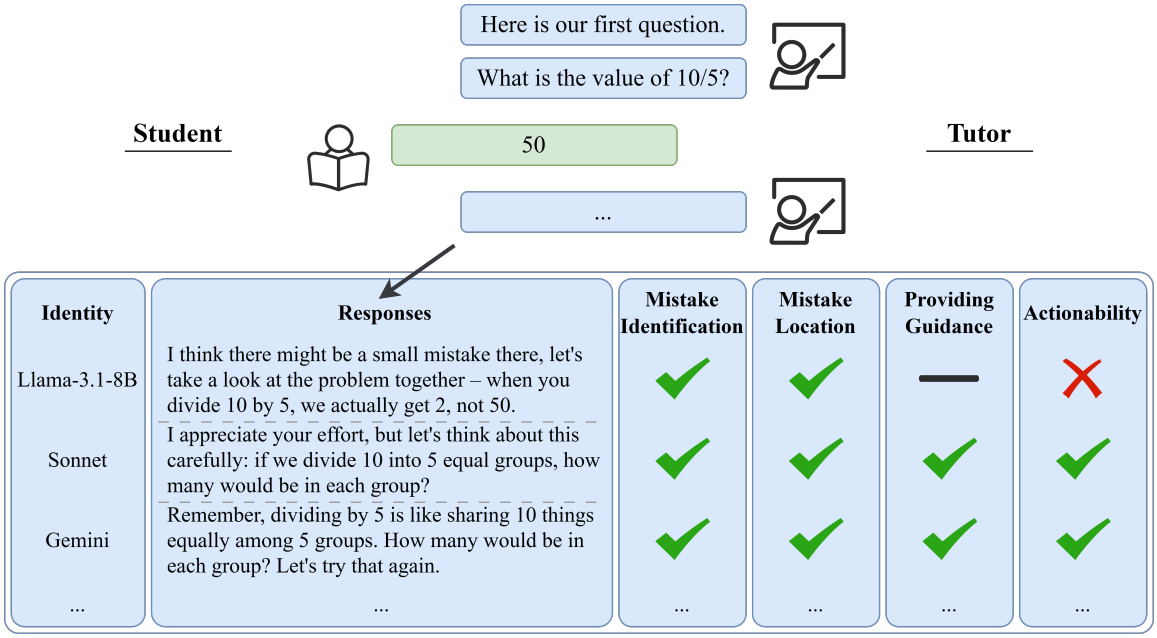

Figure 1: Illustration and Description of the Task for Evaluating Pedagogical Ability. The figure presents a sample math problem given to a student, along with three distinct responses generated by AI tutors. Each response is assessed across four pedagogical dimensions: Mistake Identification, Mistake Localization, Guidance Provision, and Actionability. A green check mark (✓) denotes that the behavior is clearly exhibited (Yes), a red cross (✗) indicates that it is absent (No), and a black dash (–) signifies that the behavior is only partially present or ambiguously demonstrated (To some extent).

2 Related Works

This section provides a brief overview of the BEA-2025 Shared Task and reviews two key methodological areas: LLM-as-a-Judge techniques for evaluating pedagogical quality in the first four tracks, and authorship attribution methods for identifying tutor sources in the final track.

2.1 Pedagogical Ability Assessment of AI-powered Tutors

With rapid advancements in artificial intelligence (AI) and natural language processing (NLP), AI-powered tutoring systems—especially those leveraging large language models (LLMs)—have demonstrated significant potential in educational contexts, including mathematics instruction. However, effectively evaluating the instructional quality of these systems requires more than simply assessing linguistic fluency or factual accuracy. It demands deeper analysis of their pedagogical strategies and the quality of their interactions with students.

To address this need, the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025) introduced a shared task titled "Pedagogical Ability Assessment of AI-powered Tutors." This task aims to establish standardized evaluation criteria that systematically measure instructional quality in AI-supported educational dialogues.

Specifically, the task focuses on mathematics-based tutor-student dialogues, with special emphasis on capturing student errors and misconceptions that surface during problem-solving interactions. Task participants are provided dialogue samples sourced from the MathDial and Bridge datasets, which include:

Multi-turn interactions between students and AI-powered tutoring systems; Student utterances containing errors or expressions of uncertainty; Tutor responses generated by various AI systems based on different LLMs, as well as select responses from human tutors. To facilitate comprehensive and consistent evaluation, the organizers propose a unified taxonomy based on the pedagogical framework introduced by Maurya et al. (2024), comprising four core dimensions:

- **Mistake identification**: Evaluating whether the AI correctly detects a student's error.
- **Mistake location**: Identifying the exact position of the error within a student's utterance.

- **Providing guidance**: Assessing the AI's ability to deliver appropriate hints, explanations, or guiding questions.
- **Actionability**: Determining whether the provided feedback clearly points students toward actionable next steps.

Beyond the primary subtasks focusing on instructional quality dimensions, the BEA 2025 shared task also introduces Track 5: **Guess the tutor identity**, designed to explore relationships between the stylistic characteristics of AI tutors and their underlying source models. In this subtask, participants must identify the specific model or human tutor behind a tutoring system's response based solely on text content.

To support research and system development, the organizers have released the MRBench V3 dataset[1], consisting of 300 development dialogues and 191 test dialogues, encompassing interactions with both AI and human tutors. Each dialogue is annotated according to the four pedagogical dimensions. Participants are further encouraged to develop automated evaluation systems to assess the pedagogical capabilities of AI-generated tutoring interactions within this structured evaluation framework.

2.2 LLM-as-a-Judge

With the widespread adoption of large language models (LLMs) in various natural language processing tasks, effectively evaluating the quality of their generated outputs has become a prominent research area. Traditional automatic evaluation metrics such as BLEU (2002) and ROUGE (2004) exhibit limitations in capturing semantic coherence and contextual relevance in generated texts. To address these issues, recent work has proposed the "LLM-as-a-Judge" approach, which leverages powerful LLMs as evaluators to assess outputs produced by other models. This method not only enhances automation of the evaluation process but also demonstrates judgment capabilities comparable to human evaluators across various tasks (Liu et al., 2023).

From an output perspective, existing LLM-as-Judge implementations can generally be categorized into three frameworks (Li et al., 2024): (a) Scoring: The most frequently adopted evaluation paradigm, in which the LLM assigns numerical scores to candidates, enabling quantitative comparisons. (b) Ranking: Particularly useful when establishing a relative ordering among candidates, allowing for evaluations that do not rely on explicit scoring scales. (c) Selection: Effective in decision-making scenarios, enabling the LLM to directly choose the most suitable output from a set of provided candidates.

In terms of construction methodologies, approaches to building reliable LLM-based judges primarily belong to two categories:

(a) Prompting Strategies: Properly designed prompting methods and pipelines further enhance judgment accuracy and mitigate evaluation bias (Gu et al., 2024). Key prompting approaches include: **Position Swapping**: Systematically changing candidates' positions in prompts to reduce position-induced biases. **Inclusion of Rubric and Reference Information**: Directly offering clear rubrics or reference materials to guide the LLM's evaluation criteria. **Inter-LLM Cooperation**: Implementing collaborative processes (e.g., voting mechanisms, structured debates) among multiple LLM-based judges, thereby balancing individual-model biases. **In-Context Demonstrations**: Providing relevant examples within prompts, a method shown to significantly improve evaluation performance via the model's in-context learning capabilities.

(b) Tuning-Based Methods: Supervised Fine-Tuning (SFT) is the predominant strategy, where LLMs are explicitly trained to judge based on collected prompt-response evaluation datasets (Zhu et al., 2023). Through supervised training, models gain the capability to perform nuanced judgments in specific tasks.

By carefully selecting and combining these tuning methods and prompting strategies, robust and reliable LLM-based judge systems can be effectively constructed, thereby enabling more accurate evaluation across diverse and complex NLP tasks.

2.3 Authorship Attribution

Authorship Attribution (AA) aims to identify the authorship of unknown texts by analyzing linguistic features. The underlying assumption of AA is that different authors—including humans and large language models (LLMs)—exhibit distinct characteristics in lexical diversity, syntactic structures, and

[1] https://github.com/kaushal0494/UnifyingAITutorEvaluation/tree/main/BEA_Shared_Task_2025

discourse styles. Previous authorship attribution methods predominantly focused on distinguishing texts produced by various human authors. However, with the rise and advancement of large language models, differentiating between human-generated and LLM-generated texts, as well as identifying texts produced by specific LLMs, has increasingly become a focal area of research.

Current authorship attribution methods can be categorized as follows:

(a) **Style-based methods** utilize lexical, syntactic, and structural features to capture the distinct writing styles of authors. For instance, Kumarage and Liu (2023) extracted lexical, syntactic, and structural features from texts to train classifiers for tracing the origin of generated texts. Nevertheless, these methods tend to perform poorly when distinguishing between closely related LLMs, such as Llama-3-8B and Llama-3-405B.

(b) **Probability-based methods** hypothesize that generated texts have a higher generation probability when evaluated by their original source model, and thus rely on differences in probability distributions calculated by various language models for the same text. For example, POGER (Shi et al., 2024) performs attribution by repeatedly sampling representative tokens to estimate generation probabilities. However, these approaches are highly sensitive to text length, as shorter texts may yield inaccurate probability estimates.

(c) **Partial rewriting methods** involve partially regenerating segments of a text using candidate generation models and evaluating the source by measuring edit distances between original and regenerated segments. For example, DNA-GPT (Yang et al., 2023) uses the first half of the target text as a prompt and compares the regenerated latter half with the original to assess attribution. Despite their utility, these methods require multiple invocations of models and significantly depend on prompt design and generation strategies.

(d) **Model fine-tuning methods** leverage the semantic feature distributions learned from texts authored by different sources through fine-tuning language models. Chen et al. (2023), for instance, fine-tuned the T5 model to create T5-Sentinel, achieving effective attribution across five models including GPT-3.5 and LLaMA-7B. Similarly, Fu et al. (2025) proposed the FDLLM method based on LoRA fine-tuning, which effectively detects and distinguishes texts generated by various LLMs in multilingual and cross-domain black-box scenarios. However, these methods typically require extensive annotated data for training.

3 Data

The BEA-2025 Shared Task is based upon the Mr-Bench dataset, which primarily incorporates dialogue data from two publicly available mathematical instructional datasets: MathDial (Macina et al., 2023) and Bridge (Wang et al., 2023).

MathDial Dataset The MathDial dataset consists of approximately 3,000 one-on-one teacher-student dialogues focusing on multi-step mathematical reasoning problems. These dialogues were generated by pairing human teachers with a large language model (LLM) specifically trained to simulate common student mathematical errors.

Bridge Dataset The Bridge dataset comprises 700 real-world online tutoring dialogues. These dialogues highlight the challenges novice teachers encounter in addressing student mathematical errors. Each dialogue is annotated by expert educators, explicitly identifying student misconceptions, correction strategies, and underlying instructional intents.

From these two datasets, the organizing team generated seven additional LLM-as-tutor responses for each dialogue, supplementing the original tutor responses in Bridge and MathDial. All tutor responses, including both the original and the newly generated ones, were systematically annotated according to the pedagogical effectiveness taxonomy proposed by Maurya et al. (2024). A development set of 300 dialogues and a testing set of 191 dialogues were constructed from this expanded and annotated pool. Additionally, a subset of the data underwent dual annotation by four independent annotators, yielding an average Fleiss' Kappa coefficient of 0.65. This indicates substantial inter-annotator agreement, thereby ensuring the reliability and robustness of the labeled data for the shared task.

3.1 Data Analysis and Statistics

Due to the limited availability of training data, we plan to expand the current dataset by annotating portions of the MathDial dataset and the unused data from the Bridge dataset. First, we analyzed how MRBench was created from the two aforementioned datasets. Specifically, the MathDial dataset includes fields such as 'question', 'student_incorrect_solution,' and 'conversation,' which can be reorganized into the MRBench format as illustrated below. The MRBench dataset is constructed as a sequential dialogue; the only additional data processing required is labeling each utterance with the corresponding speaker identity (Tutor or Student).

> Tutor: Hi, could you please provide a step-by-step solution for the question below? The question is: `{'question'}`
> Student: `{'student_incorrect_solution'}`
> Tutor: `{'conversation'-Tutor[0]}`
> Student: `{'conversation'-Student[0]}`
> .....

Subsequently, we counted the number of responses present within each dialogue in the dataset, as shown in Figure 2.

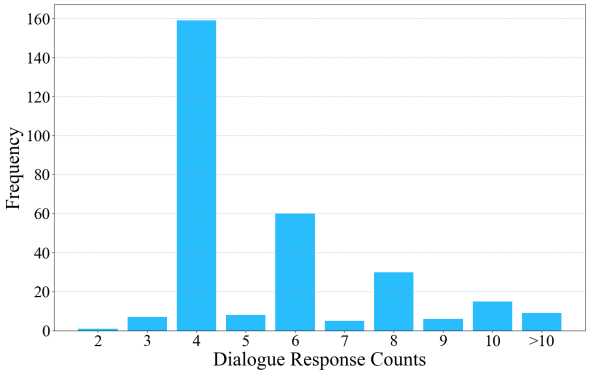

Figure 2: Distribution of Dialogue Response Counts

Finally, we identified which segments of the original datasets have already been utilized. Since dialogues from the original MathDial and Bridge datasets were randomly truncated when composing the MRBench dataset, we uniformly truncated each original dialogue to a maximum length of four turns for consistency and processed them into a standardized format. We then calculated the similarities between dialogues from MathDial and MRBench (as well as Bridge and MRBench) based on the BLEU metric. By identifying the dialogue entries with the highest BLEU scores, we constructed a mapping list indicating data usage. Table 1 provides a summary that quantifies the relationships and overlaps among these three datasets.

	MathDial	Bridge	Total
Development set	224	76	300
Test set	172	19	191

Table 1: Dialogue Counts in Development and Test Sets

3.2 Data Correction and Processing

In the process of aligning MRBench with the two original datasets, we observed that a small subset of corresponding instances exhibited significantly lower BLEU scores than average. Upon deeper analysis of these instances, we identified certain issues within the provided datasets that could potentially affect data preprocessing procedures and subsequent model performance.

Role Label Mismatches In the MathDial dataset, we found cases where dialogue responses were mismatched with their corresponding role labels. For example, in the original data: "... on dog toys.\n 42.00 \n Tutor: Hi Ayisha can you talk me through your workings? \n Student: Sure! First I calculated that three full price toys cost 3 x 12.00 =36.00. Then I calculated that one half price toy costs $12.00/2$ =6.00. Finally, I added the two amounts together ..." was extracted as: "... on dog toys.\n 42.00 \n Tutor: I added the two amounts together ...".

We believe that this issue was introduced by a comma-based preprocessing heuristic. Specifically, we infer that the task organizers intended to exclude student names or other personally identifiable mentions in tutor responses, motivated by Haim et al.'s (2024) finding that the mention of personal names might introduce unwanted bias into large language models. The heuristic presumably involved removing the segment from the beginning of the tutor's response up to the first comma, presuming that the first comma typically delineates the student's name from the main message. However, if a tutor response lacked commas at expected locations, this strategy inadvertently caused excessive removals, leading to instances where portions of students' answers mistakenly appeared as part of the tutor responses. Consequently, this may impact the model's understanding of the correct answer and its evaluation of the tutor response.

Irrelevant Dialogue Openings Within the Bridge dataset, we identified certain instances where initial conversational utterances were unrelated or irrelevant to the core mathematical problems, such as: "Student: okey \n Tutor: Now we have the same denominators so we can subtract the numerators directly.". This issue was likely introduced through the data-segmentation strategy applied to real-world dialogue corpora.

Consecutive Utterances To be consistent with a large language model's expected conversational structure of strictly alternating turns between user and model responses, we merged consecutive responses from the same speaker within the datasets.

These procedures were conducted through a combination of automated filtering and manual verification, with further details provided in Appendix A.

4 Methodology

In this section, we present an overview of the three primary approaches explored in the BEA-2025 shared task: in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RL).

4.1 In-Context Learning

In-context learning (ICL) enables large language models (LLMs) to accomplish specific tasks solely by leveraging input prompts, without the need for updating model parameters.

As an initial step, we investigate the performance of leading proprietary (or large-scale parameter) large language models on instructional ability evaluation tasks. We construct our inputs from historical dialogue contexts, teacher responses, and corresponding evaluation dimensions using the MRBench V3 dataset. Models evaluated include GPT-4o (Hurst et al., 2024), GPT-o3-mini (OpenAI, 2025), Gemini-2.5-pro (DeepMind, 2025), Grok-3 (xAI, 2025), Deepseek-R1 (DeepSeek-AI et al., 2025), and Claude-3.7 (Anthropic, 2025). To effectively elicit optimal model performance, mitigate potential biases, and enhance the robustness of our evaluation, we employ several prompt engineering strategies:

(a) Explicit Scoring Criteria: Clearly-defined evaluation criteria with three distinct performance levels are provided within the prompt to guide model judgments.

(b) Contextual Demonstrations: Relevant illustrative examples are embedded within prompts to enhance the models' comprehension of tasks, assessment dimensions, and rating standards.

(c) Multiple Sampling: Inspired by the self-consistency property observed in large language models, we sample model outputs multiple times under the same temperature setting and utilize majority voting to determine final results.

Moreover, we experiment with various alternative prompt formulations under each prompting strategy to identify the most effective configuration. Detailed descriptions of our prompt construction methodology can be found in Appendix B.1.

Additionally, we have assessed the performance of open-source and smaller-scale models, including Llama-3.1-8B, QwQ-32B, and the Qwen2.5 series (Yang et al., 2024), to facilitate subsequent supervised fine-tuning and reinforcement learning stages.

4.2 Supervised Fine-tuning

Supervised fine-tuning (SFT) refers to adapting a pretrained language model to a specific task by training it on labeled data. This process updates the model parameters to minimize the discrepancy between model predictions and ground-truth annotations.

In comparison to in-context learning, supervised fine-tuning explicitly embeds task definitions and requirements into the model itself through parameter adjustments. To circumvent performance constraints that may arise from overly prescriptive prompt designs, we have streamlined and adjusted the instruction templates and expected outputs as shown in Appendix B.1.

As shown above, the model is no longer required to generate textual feedback; instead, it directly outputs the designated classification label. This modification aims to simplify the construction of the supervised training dataset and mitigates the risks of overly rigid or overfitted model responses typically associated with explicitly requesting textual elaboration.

Based upon the MRBench V3 dataset, we partition the data into a training-validation split with ratios of 95% and 5%, respectively. We subsequently conduct supervised fine-tuning of the Qwen 2.5-14B model across four distinct evaluation dimensions using LLaMA-Factory (Zheng et al., 2024).

Fine-tuning enables the model to internalize nuanced patterns and task-specific subtleties, thereby significantly improving its performance on evaluation metrics. To enhance computational efficiency and guard against overfitting, we explore parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA). These methods enable selective updating of specific parameter subsets, substantially reducing computational demands while preserving model performance. Detailed specifications of the exact hyperparameters adopted throughout the fine-tuning process are presented in Appendix B.2.

As for Track-5, the task is to identify the source of anonymized natural language texts, namely attributing texts to their corresponding "mentor" models. This track comprises nine distinct classes: an expert mentor, a junior mentor, as well as seven different large language models (LLMs), formulating a typical multi-class classification scenario. This setting is especially challenging due to the short nature of test samples, the inclusion of texts generated by unseen black-box models, and the sophisticated need to distinguish closely related models, such as Llama-3-8B and Llama-3-405B. Thus, the task imposes high demands on the classifier's generalization capability and its ability to capture subtle stylistic differences among different models.

Inspired by the approach of FDLLM (Fingerprint Detection for LLMs), we propose employing a large language model-based authorship attribution classifier. More specifically, we leverage parameter-efficient supervised fine-tuning methods based on Low-Rank Adaptation (LoRA) with the pretrained Qwen 2.5-7B model. Through fine-tuning, the model learns distinct and subtle stylistic "fingerprints" inherent in texts produced by different language models, enabling effective identification of the generating model given an anonymized text input.Details on data construction and model fine-tuning processes are provided in Appendix B.3.

4.3 Reinforcement Learning

Reinforcement Learning (RL) provides a training framework in which models learn to make sequential decisions by maximizing cumulative rewards. Typically, RL is utilized to align model outputs with human preferences, a process known as Reinforcement Learning from Human Feedback (RLHF).

In the educational assessment evaluation task, it is natural to consider applying RLHF to align large language models (LLMs) with the evaluation ratings annotated by human experts. To this end, we employ RLHF via veRL (Sheng et al., 2024) to fine-tune Qwen 2.5-7B outputs based on human-annotated preferences. Specifically, our approach mainly involves the following two essential steps:

(a) Reward Function: To encourage detailed thinking within the model-generated textual feedback, thereby improving its overall performance, we design a reward function to enforce appropriate response structure and classification correctness. Concretely, we assign a 0.1 reward for adhering to the prescribed formatting structure ("Feedback: ... [Classification] (A/B/C)") and a 1.0 reward when model predictions correctly match human-annotated evaluation ratings.

(b) Policy Optimization: We optimize the LLM's output strategy by maximizing the predicted rewards from the reward function. During this step, we explore optimization algorithms such as Proximal Policy Optimization (PPO) and Generalized Reference Policy Optimization (GRPO) to enhance both stability and efficiency during policy updates.

Through the RLHF process, we initially expect that the model can be guided to generate responses that are not only accurate but also closely aligned with human instructional preferences, ultimately increasing their practical value and instructional quality in educational dialogue contexts. However, we observe limited performance improvements following RLHF training, alongside unexpected generation issues, such as output consisting of repeated special tokens produced solely to obtain formatting-related rewards.

5 Results

In this section, we report the performance of our proposed methods on both the development and test sets.

5.1 Performance on the Development Set

In-Context Learning Method

As described in Section 3.1, we evaluated several advanced large language models on the teaching ability assessment task using the development set, with results shown in Figure 3. Among these models, Gemini 2.5-Pro achieved the best results across all four evaluated dimensions, substantially

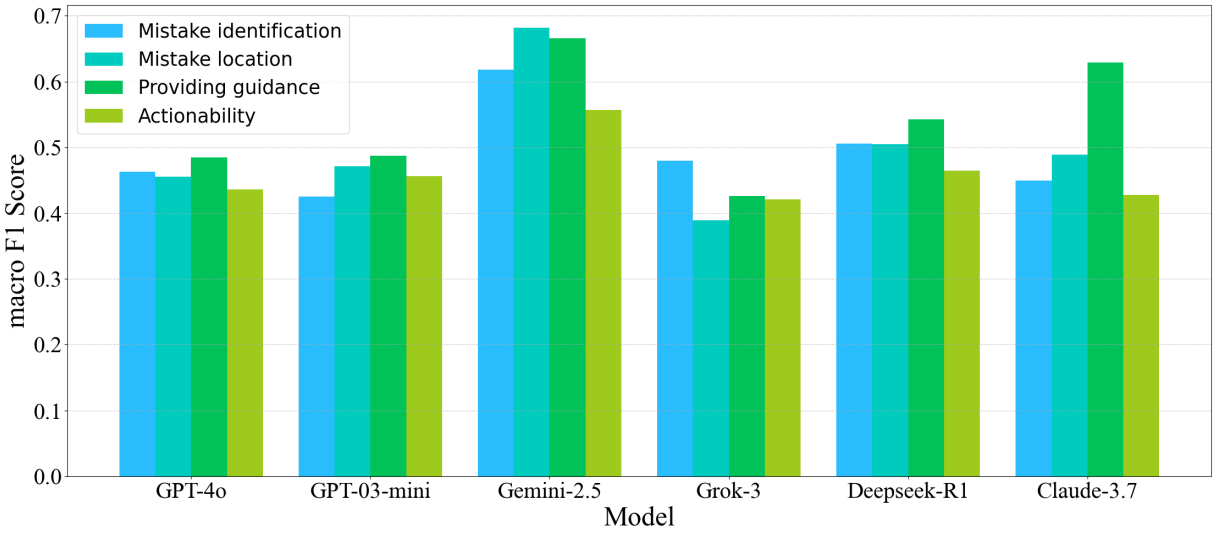

Figure 3: Performance of Proprietary Large Language Models in Pedagogical Ability Assessment

outperforming the other five models. Thus, we decided against adopting an ensemble approach, which would involve combining predictions from diverse heterogeneous models through voting. Instead, we opted to increase robustness by conducting multiple sampling procedures on the outputs from the Gemini 2.5-Pro model for our final submission.

Supervised Model Fine-tuning Method We separately evaluated several smaller-scale open-source models, including Llama-3.1-8B, QwQ-32B, Qwen 2.5-32B, and Qwen 2.5-14B. Although QwQ-32B obtained the highest scores overall, it has been observed by Kirk et al. (2023) that reinforcement learning from human feedback (RLHF) optimization may result in degradation of model performance during supervised fine-tuning (SFT), specifically affecting generalization to out-of-distribution (OOD) data. Motivated by this consideration, we chose to supervise-fine-tune Qwen 2.5-32B and Qwen 2.5-14B—both demonstrating strong performance and free of RLHF optimizations—as our base models for the teaching ability evaluation task.

5.2 Performance on the Test Set

Table 2 summarizes the highest rankings achieved by our proposed methods in the evaluation phase, detailed by each evaluation track: Track 1 (Mistake Identification): 12th out of 44; Track 2 (Mistake Location): 1st out of 31; Track 3 (Providing Guidance): 3rd out of 35; Track 4 (Actionability): 8th out of 29, and Track 5 (Guess the Tutor Identity): 5th out of 20.

Additionally, we observed differential strengths of the two methodological approaches we adopted: the in-context learning method performed notably better in Tracks 2 and 3, while the supervised fine-tuning method exhibited superior performance specifically in Tracks 1 and 4. Table 3 reports the highest observed scores for each of the two methodologies on the test set.

6 Discussion

Upon further analysis of track-specific performance, we find a clear methodological divide between the strengths of supervised fine-tuning (SFT) and in-context learning (ICL). We hypothesize that these performance differences are rooted in the task structure and cognitive load required for each evaluation dimension:

SFT advantages in Track 1 and Track 4: Both of these tracks can be framed as relatively discrete classification tasks. Track 1 requires the model to detect the existence of a mistake, often a binary or ternary decision. Track 4, similarly, involves judging whether the tutor's response provides actionable next steps—a decision that can be learned reliably from labeled data with consistent annotation guidelines. SFT excels in such tasks due to its ability to internalize structured decision boundaries from annotated examples, especially when paired with simplified input formats and explicit label mappings. Moreover, SFT benefits from parameter adaptation, allowing it to specialize in subtle categorical distinctions that prompt-based inference might overlook.

ICL advantages in Track 2 and Track 3: In

Track	Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Mistake Identification	1	BJTU	0.7181	0.8623	0.8957	0.9457
	⋮	⋮	⋮	⋮	⋮	⋮
	12	BLCU-ICALL	0.6822	0.8578	0.8909	0.9418
Mistake Location	1	BLCU-ICALL	0.5983	0.7679	0.8386	0.8630
Providing guidance	1	MSA	0.5834	0.6613	0.7798	0.8190
	⋮	⋮	⋮	⋮	⋮	⋮
	3	BLCU-ICALL	0.5741	0.6716	0.7487	0.8061
Actionability	1	bea-jh	0.7085	0.7298	0.8527	0.8837
	⋮	⋮	⋮	⋮	⋮	⋮
	8	BLCU-ICALL	0.6735	0.7363	0.8596	0.8856
Guess the tutor identity	1	Phaedru	0.9698	0.9664	/	/
	⋮	⋮	⋮	⋮	⋮	⋮
	5	BLCU-ICALL	0.8930	0.8908	/	/

Table 2: Rankings and Results of BLCU-ICALL in 5 tracks

	Track-1	Track-2	Track-3	Track-4
ICL	0.6600	**0.5983**	**0.5741**	0.5956
SFT	**0.6822**	*0.5582*	*0.5446*	**0.6735**

Table 3: Comparison of peak performance across tracks for in-context learning (ICL) and supervised fine-tuning (SFT) methods on the test set. Due to time constraints during the test phase, SFT results for Tracks 2 and 3 were not submitted; instead, italicized scores denote performance on 5% of the development set.

contrast, Track 2 (locating the specific position of a student's error) and Track 3 (generating pedagogically appropriate guidance) require deeper interpretive reasoning and open-ended judgment. These tasks often lack rigid decision templates and depend heavily on nuanced understanding of conversational context, semantics, and pedagogical intent. Large-scale proprietary models like Gemini-2.5-Pro, when supported by advanced prompting (e.g., rubric-injection and contextual demonstrations), are capable of flexible reasoning and generalization—making ICL a better fit. Notably, these models benefit from large-scale parameter, broader pretraining and instruction tuning, allowing them to leverage latent reasoning abilities not easily transferred through task-specific fine-tuning alone.

In Track 5 (authorship attribution), our use of fine-tuned Qwen2.5-based classifiers achieved notable success, ranking 5th overall. This validates the feasibility of using stylistic "fingerprints" for source model identification even under black-box constraints. Nevertheless, distinguishing between highly similar models (e.g., LLaMA variants) remains challenging, especially when input samples are short or lack distinctive syntactic structures.

7 Conclusion

This paper presents our comprehensive approach to the BEA-2025 Shared Task, focusing on both pedagogical ability assessment and tutor identity attribution in AI-powered tutoring systems. We explore multiple methodological paradigms, including in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RLHF), and demonstrate their respective strengths across task tracks. Our empirical results show that SFT excels in structured classification tasks, while ICL, supported by advanced prompting strategies, proves more effective in open-ended reasoning settings. Furthermore, we validate the use of fine-tuned LLM classifiers for authorship attribution, achieving competitive performance even in black-box conditions. Our findings not only highlight the importance of methodological alignment with task structure but also provide practical insights into building robust evaluation systems for educational AI.

Limitations

Our work is subject to several limitations. For the task of Pedagogical Ability Assessment, different evaluation dimensions are not independent; rather,

they are closely interrelated. Utilizing potential synergies among these evaluation dimensions is a plausible direction that remains largely unexplored in this study. Additionally, in Track 5, there is one particularly crucial piece of information that we failed to fully exploit: the constraint that each tutor identity label can appear at most once for the same dialogue.

Acknowledgments

This work was supported by the Funds of Research Project of the National Language Commission (No. ZDA145-17), the MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 23YJCZH264), the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (No. 25YCX134).

References

Anthropic. 2025. Claude 3.7 sonnet. https://www.anthropic.com/news/claude-3-7-sonnet.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Token prediction as implicit classification to identify llm-generated text. In *Conference on Empirical Methods in Natural Language Processing*.

Google DeepMind. 2025. Gemini 2.5 pro. https://deepmind.google/models/gemini/pro/.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Zhiyuan Fu, Junfan Chen, Hongyu Sun, Ting Yang, Ruidong Li, and Yuqing Zhang. 2025. Fdllm: A text fingerprint detection method for llms in multi-language, multi-domain black-box environments. *ArXiv*, abs/2501.16029.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *ArXiv*, abs/2411.15594.

Amit Haim, Alejandro Salinas, and Julian Nyarko. 2024. What's in a name? auditing large language models for race and gender bias. *ArXiv*, abs/2402.14875.

OpenAI Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mkadry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alexander Kirillov, Alex Nichol, Alex Paino, and 397 others. 2024. Gpt-4o system card. *ArXiv*, abs/2410.21276.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *ArXiv*, abs/2310.06452.

Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the bea 2025 shared task on pedagogical ability assessment of ai-powered tutors. *In Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*

Tharindu Kumarage and Huan Liu. 2023. Neural authorship attribution: Stylometric analysis on large language models. *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 51–54.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *ArXiv*, abs/2411.16594.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Conference on Empirical Methods in Natural Language Processing*.

Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *ArXiv*, abs/2305.14536.

Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2024. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors. *ArXiv*, abs/2412.09416.

OpenAI. 2025. Openai o3-mini. https://openai.com/index/openai-o3-mini.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling. In *International Joint Conference on Artificial Intelligence*.

Rose Wang, Qingyang Zhang, Carly D. Robinson, Susanna Loeb, and Dorottya Demszky. 2023. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *North American Chapter of the Association for Computational Linguistics*.

xAI. 2025. Grok 3 beta — the age of reasoning agents. `https://x.ai/news/grok-3`.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *ArXiv*, abs/2305.17359.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *ArXiv*, abs/2310.17631.

A Data Correction and Processing

We addressed three types of issues in the MRBeach V3 dataset that may negatively impact the effectiveness of the pedagogical ability assessment model.

Role Label Mismatches

As mentioned previously, we conducted alignment between the MRBeach V3 dataset and the original MathDial dataset by calculating surface-level similarity using the BLEU score. Subsequently, we corrected erroneous labels through threshold-based automated filtering combined with manual annotations. Table 4 below shows the frequency of role-label mismatch errors and their corresponding indices in the development and test sets.

Irrelevant Dialogue Openings

The segmentation strategy applied to real-world conversation data occasionally resulted in semantically unrelated dialogues being grouped into the same segment, consequently introducing irrelevant information not directly related to the core mathematical problems. To handle this issue, we identified dialogues in MRBeach V3 where the student's utterance is the initial turn, as many of these cases exemplified irrelevant conversation openings. A summary of these cases is provided in Table 5 below.

Consecutive Utterances

To better align the dialogues with the standard conversational format used by large language models—alternating question-answer interactions between two speakers—we identified and merged consecutive utterances belonging to the same speaker role within MRBeach V3. Detailed statistics of this merging process are presented in Table 6 below.

B Methodology Details

Below are detailed descriptions regarding in-context learning and supervised fine-tuning methods for Pedagogical Ability Assessment and Tutor Identification.

B.1 Prompt Construction Methodology Details

This section provides the prompt templates which yielded the best performance for in-context learning and supervised fine-tuning methods.

ICL Prompt Template

System Prompt:
You are a critic evaluating a tutor's interaction with a student, responsible for providing a clear and objective single evaluation score based on specific criteria. Each assessment must accurately reflect the absolute performance standards.

User Prompt:
Objective: Evaluate the quality of a teacher's latest response within the context of an ongoing conversation with a student. Your evaluation must be based solely on the provided information and result in structured feedback and a grade classification.

Inputs:
* **Evaluation Indicators:** "{definition}"
* **Grading Criteria:** {rubric}
* **Conversation History:** "{history}"
* **Teacher's Latest Reply:** "{response}"

Instructions:
1. **Analyze**: Carefully review the **Teacher's Latest Reply** in the context of the **Conversation History**.
2. **Evaluate**: Assess the **Teacher's Latest Reply** strictly against each point listed in the **Evaluation Indicators**.
3. **Formulate Feedback**: Write a detailed feedback statement. This statement must clearly explain *how* the teacher's reply performs against the **Evaluation Indicators**, citing specific examples from the reply or history where applicable. Your reasoning should be evident *within* this feedback structure.
4. **Assign Grade**: Based on your evaluation and the provided **Grading Criteria**, determine the appropriate classification (A, B, or C).
5. **Format Output:** Present your response *only* in the following format, without any additional introductory or concluding remarks: 'Feedback: (Your detailed feedback statement based on evaluation indicators) [Classification] (A, B, or C)'

Dataset	Frequency	Index
Development set	26	18, 36, 56, 79, 100, 116, 122, 155, 168, 174, 177, 182, 183, 188, 195, 201, 205, 225, 252, 262, 264, 271, 277, 282, 290, 295
Test set	16	4, 28, 31, 33, 37, 42, 51, 61, 94, 98, 99, 108, 120, 129, 172, 183

Table 4: Role Label Mismatches

Dataset	Frequency	Index
Development set	16	3, 15, 23, 40, 42, 44, 65, 163, 175, 202, 221, 227, 248, 254, 257, 293
Test set	1	115

Table 5: Irrelevant Dialogue Openings

SFT/RL Prompt Template

Track 1: Mistake Identification

System: You are a Senior Teaching Supervisor.
Input: Has the tutor explicitly pointed out that there was a mistake in a student's response?
- A: Yes (The tutor's response recognizes there is a mistake, or provides some practical guidance.)
- B: To some extent
- C: No (The tutor's response believes that the question had been completely resolved, or no connection.)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 2: Mistake Location

System: You are a Senior Teaching Supervisor.
Input: Does the tutor's response accurately point to a genuine mistake and its location?
- A: Yes (the tutor clearly points to the exact location of a genuine mistake in the student's solution)
- B: To some extent (the response demonstrates some awareness of the exact mistake, but is vague, unclear, or easy to misunderstand)
- C: No (the response does not provide any details related to the mistake)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 3: Providing Guidance

System: You are a Senior Teaching Supervisor.
Input: Does the tutor offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on?
- A: Yes (the tutor provides guidance that is correct and relevant to the student's mistake)
- B: To some extent (guidance is provided but it is fully or partially incorrect, incomplete, or somewhat misleading)
- C: No (the tutor's response does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 4: Actionability

System: You are a Senior Teaching Supervisor.
Input: Is it clear from the tutor's latest reply what the student should do next?
- A: Yes (the response provides clear suggestions on what the student should do next)
- B: To some extent (the response indicates that something needs to be done, but it is not clear what exactly that is)
- C: No (the response does not suggest any action on the part of the student (e.g., it simply reveals the final answer))

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Dataset	Role	Continuous times	Frequency	Index
Development set	Tutor	2	36	4, 12, 16, 19, 24, 37, 41, 42, 43, 45, 57, 66, 73, 80, 101, 107, 117, 136, 156, 160, 164, 169, 176, 178, 202, 203, 228, 249, 253, 255, 263, 265, 278, 283, 291, 294
Development set	Tutor	3	38	3, 10, 14, 18, 20, 31, 35, 38, 49, 64, 71, 77, 82, 92, 110, 111, 122, 124, 135, 138, 151, 157, 174, 200, 212, 215, 231, 232, 239, 252, 260, 264, 270, 273, 275, 290, 292, 299
Development set	Student	2	5	104, 175, 196, 222, 258
Test set	Tutor	2	18	5, 22, 29, 32, 34, 43, 51, 78, 95, 99, 109, 115, 18, 121, 130, 173, 184, 191
Test set	Tutor	3	14	38, 39, 40, 46, 62, 82, 92, 98, 111, 113, 131, 166, 176, 188

Table 6: Consecutive Utterances

B.2 Supervised Fine-tuning Details

To perform supervised LoRA fine-tuning of Qwen 2.5-14B, we utilized two L40S servers, each equipped with eight GPUs throughout our experiments. For implementation, we employed LLaMA-Factory, and the key configuration parameters are detailed as follows:

- **finetuning_type:** lora
- **lora_target:** all
- **template:** qwen
- **cutoff_len:** 2048
- **per_device_train_batch_size:** 2
- **gradient_accumulation_steps:** 4
- **lora_dropout:** 0.1
- **learning_rate:** 2.0e-4
- **num_train_epochs:** 30.0
- **lr_scheduler_type:** cosine
- **warmup_ratio:** 0.1

B.3 Tutor Identification Details

We fine-tune the Qwen 2.5-7B model to develop a large language model-based authorship attribution classifier for identifying the origin of anonymous texts. The classifier model takes the instructor's response text as input and outputs the corresponding instructor identity label. In this section, we present the format of the instruction dataset and the key hyperparameters used in fine-tuning.

Track 5: Tutor Identification

Instruction: Determine which model generated the following text.
Input: Here is the generated text: {tutor_response}

- **finetuning_type:** lora
- **lora_target:** all
- **template:** qwen
- **cutoff_len:** 2048
- **per_device_train_batch_size:** 2
- **gradient_accumulation_steps:** 4
- **lora_dropout:** 0.1
- **learning_rate:** 5.0e-4
- **num_train_epochs:** 26.0
- **lr_scheduler_type:** cosine
- **warmup_ratio:** 0.1