Beyond Passive Critical Thinking: Fostering Proactive Questioning to Enhance Human-AI Collaboration

Ante Wang[1, 2*], **Yujie Lin**[1, 2*], **Jingyao Liu**[2, 3*], **Suhang Wu**[3], **Hao Liu**[4], **Xinyan Xiao**[4], **Jinsong Su**[1, 2, 5†]

[1]School of Informatics, Xiamen University, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
[3]Department of Digital Media Technology, Xiamen University, China
[4]Baidu Inc., Beijing, China
[5]Shanghai Artificial Intelligence Laboratory, China
{wangante,yjlin,liujingyao}@stu.xmu.edu.cn, jssu@xmu.edu.cn

Abstract

Critical thinking is essential for building robust AI systems, preventing them from blindly accepting flawed data or biased reasoning. However, prior work has primarily focused on passive critical thinking, where models simply reject problematic queries without taking constructive steps to address user requests. In this work, we introduce proactive critical thinking, a paradigm where models actively seek missing or clarifying information from users to resolve their queries better. To evaluate this capability, we present GSM-MC and GSM-MCE, two novel benchmarks based on GSM8K for assessing mathematical reasoning under incomplete or misleading conditions. Experiments on Qwen3 and Llama series models show that, while these models excel in traditional reasoning tasks, they struggle with proactive critical thinking, especially smaller ones. However, we demonstrate that reinforcement learning (RL) can significantly improve this ability. By incorporating heuristic information into the reward function, we achieve substantial gains, boosting the Qwen3-1.7B's accuracy from 0.15% to 73.98% on GSM-MC. We hope this work advances models that collaborate more effectively with users in problem-solving through proactive critical thinking.

Code — https://github.com/XMUDeepLIT/Learn2Ask

1 Introduction

Large Language Models (LLMs) have made significant strides in solving complex tasks, such as mathematical reasoning (Yamauchi et al. 2023; IMANI, Shrivastava, and Du 2024; Wang et al. 2025a), code generation (Feng et al. 2023; Dong et al. 2024; Shao et al. 2025; Wang et al. 2025b), and planning (Yao et al. 2023; Zhou et al. 2024; Lin et al. 2025). However, most existing studies focus on controlled settings where user queries are always answerable. In real-world applications, users, especially those without domain expertise, often provide insufficient or inaccurate information, making it difficult for models to solve problems effectively (Zamfirescu-Pereira et al. 2023; Kim et al. 2024). For instance, a patient lacking medical knowledge might omit critical symptoms, preventing an AI doctor from making a precise diagnosis (Alkaabi and Elsori 2025).

Some prior work (Rahman et al. 2024; Kirichenko et al. 2025) has acknowledged this issue, advocating for critical thinking in LLMs, which refers to the ability to reject unanswerable or flawed requests instead of attempting to process biased or incomplete inputs. Yet, we argue that this form of critical thinking remains *passive*, as it still relies on users to independently identify and rectify gaps in their queries, rather than actively facilitating problem-solving.

To address this limitation, we propose *proactive critical thinking*: a paradigm where the model not only detects unanswerable queries but also provides constructive feedback to guide users in supplying necessary information. As shown in Figure 1, this approach fosters more effective human-AI collaboration, enabling iterative conversations that progressively refine the problem and lead to a solution.

Previous research (Ma et al. 2024; Fan et al. 2025) has established some benchmarks for evaluating passive critical thinking by performing operations such as removing or replacing key information in original questions using advanced LLMs. However, these datasets often lack rigorous quality control, resulting in many noisy and overly simplistic cases (see Table 1) that are unsuitable for evaluating proactive critical thinking.

To tackle this, we develop an automated data preparation pipeline using DeepSeek-V3 (Liu et al. 2024). Following prior work (Ma et al. 2024; Sun et al. 2024), we focus on mathematical reasoning and derived our dataset from the widely used GSM8K (Cobbe et al. 2021). We identify key variables in each question, randomly remove one, and rephrase the question for fluency. This multi-step process enhances diversity and mitigates the bias introduced by the single-step methods used in previous work. To ensure quality, we apply strict filtering, retaining only ques-

*These authors contributed equally.
†Corresponding author.

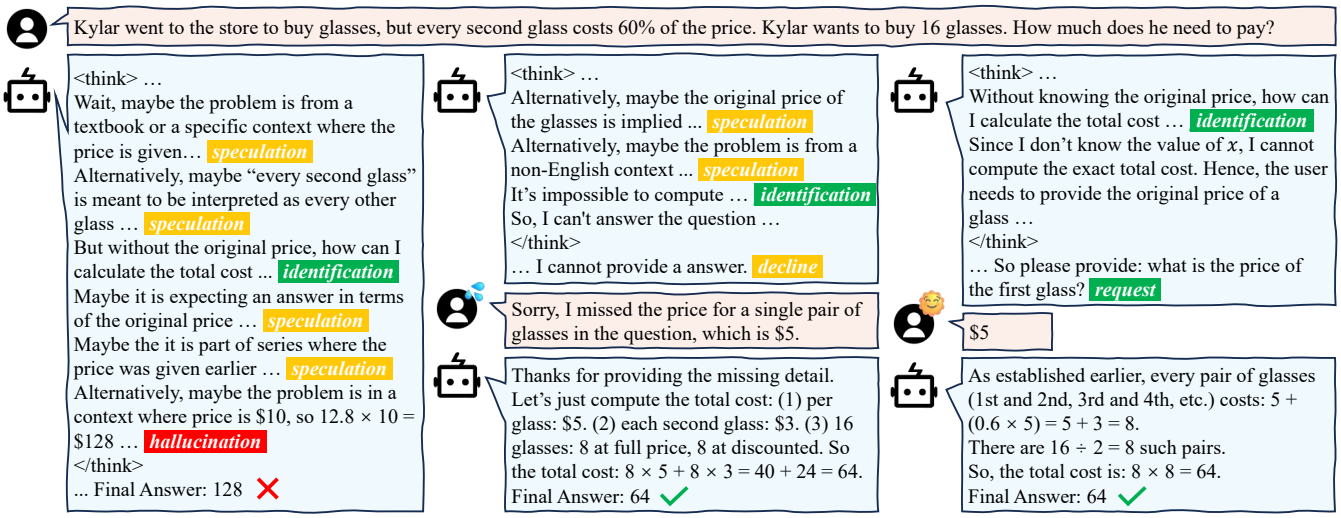

Figure 1: A comparison of model reasoning approaches to flawed questions. **Left:** Vanilla reasoning produces repetitive speculation and hallucinations, leading to an incorrect answer. **Middle:** Passive critical thinking correctly refuses to answer but shifts the burden of correction to the user. **Right:** Proactive critical thinking actively seeks clarification, greatly reducing user effort.

tions that DeepSeek-V3 can resolve through proactive critical thinking. This yields 1,368 high-quality instances, which we name GSM-MC. To increase the challenge and evaluate model robustness against distractions, we further augment these questions with irrelevant information. The resulting dataset, GSM-MCE, undergoes the same rigorous filtering, producing 1,134 instances.

To instill proactive critical thinking capabilities into models, we investigate both supervised fine-tuning (SFT) and reinforcement learning (RL). Using the data preparation pipeline described above, we combine revised unanswerable questions with original questions to construct the training set. Furthermore, we enhance both training paradigms by incorporating heuristic information about question answerability. This approach effectively increases the diversity of the SFT data and accelerates RL convergence through more dense reward signals.

We evaluate popular Qwen3 (Yang et al. 2025) and Llama (Meta 2024) series models on our GSM-MC and GSM-MCE benchmarks. The results reveal that despite extensive post-training, these models still struggle with proactive critical thinking, particularly the smaller ones. Notably, while recent inference-time scaling approaches have significantly advanced performance on complex reasoning tasks, we find that they can hinder proactive critical thinking capability. However, our training approach successfully enhances proactive critical thinking performance while maintaining accuracy on standard questions across model sizes. Most impressively, for Qwen3-1.7B, we achieve a substantial improvement in final answer accuracy from 0.15% to 73.98%. Further analysis shows that our method also generalizes effectively to out-of-distribution unanswerable questions. Our key contributions are summarized as follows:

- We introduce the concept of proactive critical thinking: a paradigm where models not only identify unanswerable queries but also provide constructive feedback to help users supply missing information.
- We develop an automated data preparation pipeline for generating high-quality flawed questions and construct two novel benchmarks, GSM-MC and GSM-MCE, for evaluating proactive critical thinking.
- We demonstrate that both SFT and RL can effectively cultivate proactive critical thinking abilities, and show that their efficacy is significantly enhanced by incorporating answerability heuristics.

2 Related Work

While LLMs are increasingly expected to handle real-world tasks, they often struggle with the ambiguous or flawed queries that users frequently pose. To address this issue, prior work (Li et al. 2024; Ma et al. 2024; Sun et al. 2024; Song, Shi, and Zhao 2025; Fan et al. 2025) has focused on investigating the critical thinking capabilities of LLMs. Most of these studies have concentrated on the mathematical domain, constructing datasets by modifying well-formed problems to make them unanswerable. Their findings show that current LLMs frequently fail to accurately detect input flaws (Ma et al. 2024), and even the latest large reasoning models (LRMs) rarely decline to answer flawed questions (Fan et al. 2025). However, Song, Shi, and Zhao (2025) demonstrates that critical thinking can be effectively improved through specific training. Nevertheless, we argue that this approach remains passive and may have limited usefulness in addressing user requests, as it still requires users to identify and correct errors themselves.

In this work, we introduce proactive critical thinking, enabling models to move beyond mere flaw detection and actively guide users with clear and targeted feedback. Existing research (Kuhn, Gal, and Farquhar 2022; Wang et al. 2024; Andukuri et al. 2024; Zhang, Knox, and Choi 2025;

Li et al. 2025) has primarily focused on asking clarifying questions in response to ambiguous user requests. However, these approaches often excel only at detecting obvious flaws, such as missing variables in tool usage (Wang et al. 2024), or operate in general conversational settings with limited ambiguity (Andukuri et al. 2024; Zhang, Knox, and Choi 2025). More complex cases requiring deeper reasoning remain under-explored. The most closely related work to ours is the recent COLLABLLM (Wu et al. 2025), which shares the same goal of enhancing human-AI collaboration through multi-turn conversations. In contrast, our work focuses on critical thinking, where an LLM should not only learn to collaborate with humans but also identify flaws and provide feedback. To this end, we construct new datasets tailored to this objective and emphasize the role of reasoning in this setting, aspects that are orthogonal to COLLABLLM.

3 Preliminary

We define *proactive critical thinking* as the ability of a model to actively collaborate with humans rather than passively refusing to respond when receiving flawed inputs.

Proactive Questioning: A Preliminary Exploration on Proactive Critical Thinking In this work, we begin by formalizing the simplest scenario: Given a question x that may lack key information, the LLM π first attempts to generate its response y through proactive critical thinking. To enable this capability, we augment the input with the following instruction:

Instruction for Activating Proactive Questioning

Question:
`[QUESTION]`

If the question is answerable, provide the final answer. Otherwise, ask the user for the necessary information by phrasing the request as a question.

If the question x is answerable, the LLM directly provides the solution $y = \pi(x)$. Otherwise, it identifies the missing information and proactively generates a follow-up query $q = \pi(x)$ to request clarification. Upon receiving the user's response a to the query q, the LLM then synthesizes the final solution $y = \pi(x, q, a)$ using all available information.

Simulating a User with a User Agent In the above setting, a user is required to respond to the LLM's request. Since it is impractical to involve human participants, we use a strong LLM to simulate the user.

For an unanswerable question x, we provide the user agent with its unmodified version, denoted as $\hat{x}$, that retains all necessary information. When the user agent receives a clarification query q, it will generate the corresponding reply a according to the original question $\hat{x}$. Due to page limitations, the user agent prompt is available in the GitHub repository.

4 Benchmarks

Following previous work (Ma et al. 2024; Sun et al. 2024), we adopt the widely used GSM8K dataset (Cobbe et al. 2021) as the foundation for constructing our benchmarks to evaluate proactive critical thinking. We introduce an automated data construction pipeline that generates and selects high-quality evaluation examples through four key steps: (1) Variable Recognition, (2) Unanswerable Question Creation, (3) Irrelevant Information Injection, and (4) Sampling-Based Filtering. All steps are implemented using DeepSeek-V3[1].

Variable Recognition For each instance, we begin by identifying all key information elements within the natural question. To maintain accurate variable positions for subsequent steps, we apply the following prompt template to annotate each variable without changing the original content. The template includes demonstration examples to guide the LLM's formatting:

Prompt for Variable Recognition

Identify and annotate key information in math questions by enclosing each piece in square brackets [].

Examples:

Input:
Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Output:
Natalia sold clips to [48 of her friends in April], and then she sold [half as many clips in May]. How many clips did Natalia sell altogether in April and May?

[Other demonstrations...]

Now, annotate the following question:
`[Question]`

Unanswerable Question Creation For each annotated question, we randomly remove one variable and instruct the LLM to rephrase it into a well-formed but unanswerable question. The following prompt is designed to ensure fluency:

Prompt for Unanswerable Question Creation

Given a question with removed information, eliminate any unnatural phrasing while maintaining fluency.

The question is: `[Question]`

Unlike previous approaches that directly prompt LLMs to edit original questions, our method reduces bias by preventing LLMs from consistently applying similar editing, thereby enhancing question diversity.

[1]DeepSeek-V3-0324

Error Type	Description	Example
Answerable Question	The question is still answerable after removing key information, or the missing detail can be inferred or corrected without ambiguity.	**Original:** The first man to walk on the moon, Neil Armstrong, was from which country? **Modified:** The first man to walk on the moon was from which country?
Obvious Information Missing	The missing information is easily identifiable due to vague terms (e.g., "some", "a certain number").	**Original:** A box contains 10 red balls and 5 blue balls. How many balls are there in total? **Modified:** A box contains some red balls and 5 blue balls. How many balls are there in total?
Unclarifiable Question	There is confusion about the appropriate perspective for making a clarification request, increasing the difficulty of evaluation.	**Original:** In triangle ABC, $\angle C = 90°$. Legs $AC = 3cm$, $BC = 4cm$. Find AB. **Modified:** In triangle ABC, $AC = 3cm$, $BC = 4cm$. Find AB.

Table 1: Summarized error types with their descriptions and typical examples. For clarity, the examples are manually constructed rather than selected from the GSM8K dataset.

Irrelevant Information Injection To increase the difficulty of problems for further challenging the model's proactive critical thinking, we introduce an optional step that adds distracting but seemingly reasonable details. The newly resulting questions allow us to test whether models can focus on relevant information while ignoring irrelevant content. The prompt utilized for this procedure is detailed below:

Prompt for Irrelevant Information Injection

Hide a misleading detail in the given question. The new question should yield the same answer as the original.

Examples:

Input:
Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Output:
Natalia sold clips to 48 of her friends in April. **She found that half of them preferred blue clips**. Then in May, she sold half as many clips as she did in April. How many clips did Natalia sell altogether in April and May?

[Other demonstrations...]

Now, hide a misleading detail in the following question:
`[Question]`

Sampling-Based Filtering While the previous steps generate diverse unanswerable questions, we still observe persistent quality issues similar to those reported in prior studies. To address this, we conduct an error analysis (see Table 1) and identify three main categories of problems. Based on these findings, we then design specific filtering strategies to mitigate them.

For each question, we sample 16 solutions generated by DeepSeek-V3 (with the user agent also implemented using DeepSeek-V3). To ensure data quality, we apply the following filters:

- **Answerable Question:** Remove questions where the LLM provides direct solutions in the first turn for more than 12 cases, indicating high confidence that the question is answerable without further interaction.
- **Obvious Information Missing:** Eliminate questions that are successfully addressed in the second turn for more than 12 cases, suggesting the question may be overly simplistic.
- **Unclarifiable Question:** Discard questions that remain unresolved in all 16 attempts, as they may be too challenging to support meaningful interaction.

After the rigorous filtering process, we obtain two high-quality datasets:

- **GSM-MC**: Contains 1,368 questions made unsolvable by removing key information.
- **GSM-MCE**: An enhanced version with 1,134 questions incorporating irrelevant information to increase difficulty.

5 Methods

In this section, we explore two approaches to enhance proactive critical thinking: traditional supervised fine-tuning (SFT) and recently prevalent reinforcement learning (RL). Additionally, we leverage heuristic information about whether a question is answerable to further improve model performance.

Supervised Fine-Tuning The most straightforward approach is to fine-tune the LLM directly on prepared human-AI interaction trajectories. Given a trajectory (x, q, a, y), we apply the standard cross-entropy loss to optimize the model π:

$$\mathcal{L}_{\text{sft}} = -\log p(q \mid x, \pi) - \log p(y \mid x, q, a, \pi). \quad (1)$$

Since no training data is readily available, we follow the same data preparation pipeline described in §4 to generate unanswerable questions using the training set of GSM8K. However, due to the high computational cost of sampling-based filtering, we instead employ a smaller LLM to perform rejection sampling and collect training trajectories.

Nevertheless, because the LLM lacks inherent proactive critical thinking capabilities, efficiently obtaining valid trajectories remains challenging. To address this, we enhance the LLM's questioning ability by incorporating heuristic information through prompting:

Instruction Enhanced with Heuristic Information

Question:
`[QUESTION]`

This question is unanswerable due to missing key information. Ask the user for the necessary information by phrasing the request as a question.

This approach encourages the LLM to seek missing information through targeted questioning, thereby improving the recall to reach correct answers. To prevent a performance decline on natural questions, we also incorporate their corresponding trajectories for training.

Reinforcement Learning On-policy RL has proven effective in enabling LLMs to independently explore strategies to achieve target objectives. In this work, we adopt the popular GRPO algorithm (Shao et al. 2024), training the model on the same question set used for SFT:

$$\mathcal{L}_{\text{GRPO}} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t} - \beta\mathcal{D}_{\text{KL}}\left[\pi_\theta\|\pi_{\text{ref}}\right]\right], \tag{2}$$

where $\hat{A}$ represents the advantage computed using outcome rewards after group normalization, and the KL divergence term prevents the policy from deviating too far from the reference policy. Here, o_i denotes a rollout interaction trajectory. Note that we exclude the loss calculation for the user-provided turns (i.e., responses to the LLM's requests) in our optimization objective.

For the reward function, we primarily consider the correctness of the final answer, assigning $r = 1$ for correct answers and $r = 0$ for incorrect ones. However, due to the sparsity of this reward signal, training may converge slowly. To mitigate this issue, we augment the reward function with the following heuristic signals:

- For answerable questions, the LLM receives a penalty of -0.5 when making unnecessary requests.
- For unanswerable questions, the LLM obtains a reward of $+0.5$ for making requests, independent of the final answer's correctness.

6 Experiments

6.1 Setup

Datasets and Evaluation Metrics We conduct experiments on the standard GSM8K (Cobbe et al. 2021) test set, and our proposed benchmarks GSM-MC and GSM-MCE (detailed in §4). Model performance is evaluated using the following metrics:

- **Accuracy (ACC)** measures the correctness of the model's answer. For GSM8K, we evaluate the initial response; for GSM-MC and GSM-MCE, where clarification is needed, we evaluate the second-turn response.
- **Request Ratio (REQ)** quantifies the model's tendency to seek missing information. Concretely, it is defined as the proportion of instances in which the model generates a clarifying question in its initial response.

Models and Implementation Details The primary experiments are conducted on a selection of representative models varying in size and architecture, including (1) the compact Qwen3-1.7B model (Yang et al. 2025) with an optional "thinking mode", (2) the larger Qwen3-8B model for assessing scalability, and (3) the Llama-3.2-3B-Instruct model (Meta 2024), from a different architecture family, to verify the generalizability of the training approach.

For SFT, we first use the Qwen3-8B model to collect complete trajectory data via rejection sampling. The models are then trained on this data for one epoch with a learning rate of 5e-6, employing a cosine learning rate scheduler with a warmup phase covering 10% of the total training steps. For RL, we adopt a learning rate of 1e-6 and set the number of rollouts to 8.

We use the Qwen3-14B model as the user agent for data construction and training to balance cost and validity, while DeepSeek-V3 (Liu et al. 2024) is employed during evaluation to ensure more accurate results.

6.2 Main Results

Vanilla models fail to provide effective feedback to flawed prompts. As presented in Table 2, off-the-shelf models struggle with proactive critical thinking when confronted with flawed or ambiguous prompts. This limitation is particularly pronounced in smaller models. For instance, Qwen3-1.7B and Llama-3.2-3B-Instruct show almost no capacity to handle these imperfect questions, with both ACC and REQ approaching 0% on GSM-MC and GSM-MCE.

Although the larger Qwen3-8B model demonstrates some ability to generate clarification requests, its performance still undergoes a significant drop. Its ACC on GSM-MC decreases by nearly 50% compared to that on the standard GSM8K benchmark, with an even sharper decline observed on GSM-MCE. This suggests that, even extensive post-training has not equipped these models with the crucial skill of actively seeking the user's help when faced with problematic queries. This shortcoming is likely attributed to a lack of exposure to noisy and imperfect training data, which more accurately mirrors the complexities and ambiguities of real-world scenarios.

Training unlocks proactive critical thinking. In a significant departure from vanilla performance, both SFT and RL yield substantial improvements in ACC and REQ. Notably, the Qwen3-1.7B and Llama-3.2-3B-Instruct models exhibit a remarkable 70% increase on GSM-MC and a 40% improvement on GSM-MCE after two-stage training. The performance gap with GSM8K is significantly narrowed.

Interestingly, the results for the Qwen3-8B model present an unexpected phenomenon: employing RL alone surpasses

Models	GSM8K		GSM-MC		GSM-MCE	
	ACC	REQ	ACC	REQ	ACC	REQ
Qwen3-1.7B w/o think						
Vanilla	78.01	0.00	0.95	1.75	0.62	1.15
SFT	78.32	6.90	38.60	77.70	17.46	69.40
RL	82.03	7.73	60.82	91.59	37.13	88.18
SFT+RL	**85.44**	6.75	**73.68**	95.69	**40.04**	90.12
Qwen3-1.7B w/ think						
Vanilla	**87.79**	0.00	0.15	0.66	0.00	0.00
SFT	86.13	0.68	44.15	70.54	14.99	44.44
RL	87.41	5.46	62.13	92.54	29.19	84.22
SFT+RL	85.75	5.91	**73.98**	97.00	**41.09**	92.95
Qwen3-8B w/o think						
Vanilla	**92.19**	0.68	55.59	76.06	25.09	59.08
SFT	91.67	1.74	65.35	91.23	27.78	81.75
RL	90.60	5.53	**81.73**	99.63	**42.59**	95.86
SFT+RL	91.21	4.55	79.24	99.12	36.16	95.33
Qwen3-8B w/ think						
Vanilla	94.62	0.08	44.92	57.75	14.11	29.59
SFT	**95.60**	0.30	57.38	75.22	21.52	48.32
RL	92.27	5.00	**85.53**	99.12	**49.38**	94.80
SFT+RL	93.48	3.26	83.11	98.61	41.62	92.86
Llama-3.2-3B-Instruct						
Vanilla	64.06	0.08	0.15	0.44	0.09	0.26
SFT	48.60	30.93	31.87	86.77	14.46	79.10
RL	**79.15**	7.96	57.97	91.37	28.92	84.74
SFT+RL	75.74	12.59	**74.49**	96.42	**48.32**	91.98

Table 2: Accuracy (ACC) and Request Ratio (REQ) of the models on the standard GSM8K and our two constructed benchmarks, GSM-MC and GSM-MCE. For the Qwen3 models, we report results both with and without the "thinking mode" to examine the role of reasoning in this task.

the performance of two-stage training. This may arise from the nature of the SFT data, which is self-generated by the Qwen3-8B model and thus does not inherently enhance its capabilities. Moreover, by further reinforcing its original high-probability tokens during SFT, the entropy of the model's outputs may be inadvertently reduced. This could constrain the exploratory nature of the subsequent RL phase, thereby hindering its overall effectiveness.

Training activates a beneficial "thinking mode". A notable observation from our experiments is that RL fundamentally changes how models use their internal "thinking mode". For vanilla models, activating the "thinking mode" often degrades performance. The extended thinking appears to induce counterproductive self-doubt rather than useful analysis, leading to a clear drop in performance. For example, the Qwen3-8B model's ACC on GSM-MC decreases from 55.59% to 44.92% when enabling the "thinking mode".

However, this trend is decisively reversed after RL training. The "thinking mode" evolves into a powerful asset, enabling the model to carefully analyze flaws in prompts and formulate effective clarifying questions. This transformation is most evident in the Qwen3-8B, where enabling the "thinking mode" increases its ACC on GSM-MC from 81.73% to a peak of 85.53%. Additionally, we observe that RL also enhances the model's thinking ability even without explicitly activating the "thinking mode", as evidenced by the increasing response length.

Models	GSM8K		GSM-MC		GSM-MCE	
	ACC	REQ	ACC	REQ	ACC	REQ
Qwen3-1.7B w/o think						
SFT	78.32	6.90	38.60	77.70	17.46	69.40
w/o filter	76.65	5.23	34.71	68.93	14.98	57.93
RL	82.03	7.73	60.82	91.59	37.13	88.18
w/o filter	78.99	12.51	55.19	89.33	29.63	87.13
Qwen3-1.7B w/ think						
SFT	86.13	0.68	44.15	70.54	14.99	44.44
w/o filter	90.75	0.38	27.20	48.25	9.44	27.34
RL	87.41	5.46	62.13	92.54	29.19	84.22
w/o filter	84.46	9.78	57.09	89.62	21.96	81.92

Table 3: Results of the ablation study on the data filtering process.

General capabilities are preserved. A critical outcome of our training is that the substantial gains in proactive critical thinking do not come at the cost of the models' core mathematical reasoning abilities. As presented in Table 2, the performance of our trained models on the standard GSM8K benchmark remains remarkably stable. In fact, for smaller models, training even yields slight improvements in their foundational skills. For instance, the Llama-3.2-3B-Instruct model's ACC on GSM8K increases significantly from 64.06% to 75.74% after undergoing the full SFT+RL training. This demonstrates that our approach successfully integrates a new, specialized skill of proactive critical thinking, while simultaneously preserving and even enhancing the models' general capabilities, resulting in more robust and reliable models fit for complex, real-world applications.

6.3 Analysis

The Importance of Data Filtering To verify the filtering process in our data curation method, we conduct an ablation study on Qwen3-1.7B model. In this experiment, we train a control version of the model using an unfiltered dataset created by randomly sampling from the original data pool. The size of this unfiltered dataset is kept identical to our curated, high-quality dataset to ensure a fair comparison. As detailed in Table 3, removing the data filtering pipeline leads to a significant degradation in model performance. For instance, when the "thinking mode" is enabled, the ACC of the SFT model on the GSM-MC benchmark falls sharply from 44.15% to just 27.20%. This substantial decrease underscores the criticality of our carefully designed data filtering process in enabling the model to acquire proactive critical thinking skills during training.

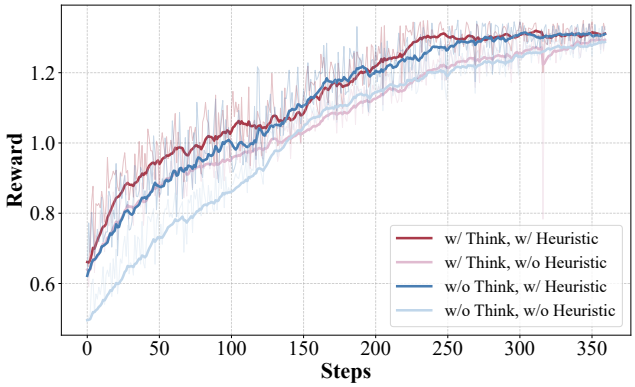

Figure 2: Reward curves in the RL stage following SFT, comparing the reward progression with and without incorporating heuristic information.

Methods	Qwen3-8B w/o think		Qwen3-8B w/ think	
	ACC	REQ	ACC	REQ
Vanilla	21.15	32.69	17.31	28.84
SFT	42.31	73.08	26.92	46.15
RL	**57.69**	80.77	**59.62**	76.92
SFT+RL	50.00	80.77	55.77	78.85

Table 4: Qwen3-8B model's performance on the out-of-distribution MIP-MATH dataset.

The Role of Heuristic Guidance To evaluate the influence of heuristic guidance during training, we analyze the reward curves in the RL phase. As shown in Figure 2, models trained with heuristics (the red and blue lines) begin with a substantially higher initial reward compared to their counterparts trained without this guidance (the pink and light-blue lines). Furthermore, they exhibit a markedly faster rate of convergence, thereby achieving higher performance sooner. This distinct advantage persists regardless of whether the "thinking mode" is enabled, demonstrating that heuristic guidance serves as a powerful catalyst for more efficient RL.

Generalization Analysis To assess the generalization capabilities of our trained models, we conduct an out-of-distribution evaluation using the MIP-MATH dataset (Fan et al. 2025). This challenging benchmark is derived from the standard MATH dataset (Hendrycks et al. 2021) by intentionally omitting a critical premise from each problem manually. The results, presented in Table 4, demonstrate that our trained Qwen3-8B models significantly outperform their vanilla counterparts, irrespective of whether the "thinking mode" is enabled. This robust performance on the challenging, out-of-distribution benchmark indicates that the proactive critical thinking skills imparted by our method are not confined to the training distribution. Instead, our approach fosters a generalizable capability, further validating the effectiveness of our data curation and training paradigm.

Multi-Turn Expansion Experiments Finally, we investigate whether the proactive critical thinking skills instilled in our models can generalize to a multi-turn dialogue setting, a scenario that is not explicitly included in training. To evaluate this, we design an experiment where the model must iteratively assess whether it has gathered sufficient information to answer, engaging in multiple rounds of clarification if necessary. The results, detailed in Table 5, reveal a remarkable trend: model performance consistently improves as the number of interactive turns increases. For instance, the RL-trained Qwen3-8B model (with thinking enabled) improves its ACC from 85.53% in a standard two-turn interaction to 88.67% after four dialogue turns. This steady enhancement indicates that the learned proactive critical thinking is not a rigid, single-step capability but rather an adaptable skill that naturally extends to more complex interactions.

Turns	Qwen3-1.7B		Qwen3-8B	
	w/o think	w/ think	w/o think	w/ think
SFT				
2	38.60	44.15	65.35	57.38
3	43.49	49.71	66.01	57.73
4	**45.69**	**50.64**	**66.52**	**58.12**
RL				
2	60.82	62.13	81.73	85.53
3	61.26	62.72	83.26	88.30
4	**61.99**	**63.74**	**84.06**	**88.67**
SFT+RL				
2	73.68	73.98	79.24	83.11
3	74.63	74.49	80.48	85.96
4	**74.93**	**75.23**	**81.07**	**86.42**

Table 5: The ACC on the GSM-MC benchmark across multiple conversational turns.

7 Conclusion

This work presents a preliminary study on proactive critical thinking in LLMs, where the model not only identifies flaws in user input but also actively engages with users to collaboratively solve problems. We first propose a data preparation pipeline to construct GSM-MC and its more challenging extension, GSM-MCE, enabling systematic evaluation of proactive critical thinking. Building on these benchmarks, we then improve model performance through supervised fine-tuning and reinforcement learning, enhanced with heuristic guidance. Experimental results demonstrate that our approach leads to significant improvements on both test sets across three different models. As this area is still underexplored, several promising directions remain: (1) Developing benchmarks covering a broader range of domains, such as medicine. (2) Extending interaction length to more turns to address more complex tasks. (3) Exploring more robust training algorithms capable of providing diverse feedback beyond questioning.

Acknowledgments

The project was supported by National Key R&D Program of China (No. 2022ZD0160501), Natural Science Foundation of Fujian Province of China (No. 2024J011001), and the Public Technology Service Platform Project of Xiamen (No.3502Z20231043). We also thank the reviewers for their insightful comments.

References

Alkaabi, A.; and Elsori, D. 2025. Navigating digital frontiers in UAE healthcare: A qualitative exploration of healthcare professionals' and patients' experiences with AI and telemedicine. *PLOS Digital Health*, 4(4): e0000586.

Andukuri, C.; Fränken, J.-P.; Gerstenberg, T.; and Goodman, N. D. 2024. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Dong, Y.; Jiang, X.; Jin, Z.; and Li, G. 2024. Self-collaboration code generation via chatgpt. *ACM Transactions on Software Engineering and Methodology*, 33(7): 1–38.

Fan, C.; Li, M.; Sun, L.; and Zhou, T. 2025. Missing Premise exacerbates Overthinking: Are Reasoning Models losing Critical Thinking Skill? *arXiv preprint arXiv:2504.06514*.

Feng, Y.; Vanam, S.; Cherukupally, M.; Zheng, W.; Qiu, M.; and Chen, H. 2023. Investigating code generation performance of ChatGPT with crowdsourcing social data. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 876–885. IEEE.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

IMANI, S.; Shrivastava, H.; and Du, L. 2024. Mathematical reasoning using large language models. US Patent App. 18/144,802.

Kim, Y.; Lee, J.; Kim, S.; Park, J.; and Kim, J. 2024. Understanding users' dissatisfaction with chatgpt responses: Types, resolving tactics, and the effect of knowledge level. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, 385–404.

Kirichenko, P.; Ibrahim, M.; Chaudhuri, K.; and Bell, S. J. 2025. AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. *arXiv preprint arXiv:2506.09038*.

Kuhn, L.; Gal, Y.; and Farquhar, S. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.

Li, Q.; Cui, L.; Zhao, X.; Kong, L.; and Bi, W. 2024. GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2961–2984.

Li, S. S.; Mun, J.; Brahman, F.; Ilgen, J.; Tsvetkov, Y.; and Sap, M. 2025. Aligning LLMs to Ask Good Questions A Case Study in Clinical Reasoning. *CoRR*, abs/2502.14860.

Lin, Y.; Wang, A.; Chen, M.; Liu, J.; Liu, H.; Su, J.; and Xiao, X. 2025. Investigating Inference-time Scaling for Chain of Multi-modal Thought: A Preliminary Study. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 15654–15667. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Ma, J.; Dai, D.; Yuan, Z.; Luo, W.; Wang, B.; Liu, Q.; Sha, L.; Sui, Z.; et al. 2024. Large Language Models Struggle with Unreasonability in Math Problems. *arXiv preprint arXiv:2403.19346*.

Meta, A. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI Blog. Retrieved December*, 20: 2024.

Rahman, A.; Ye, J.; Yao, W.; Liu, S. S.; Yu, J.; Yu, J.; Yin, W.; and Wang, G. 2024. From Blind Solvers to Logical Thinkers: Benchmarking LLMs' Logical Interity on Faulty Mathematical Problems. *arXiv preprint arXiv:2410.18921*.

Shao, L.; Yan, Y.; Poshyvanyk, D.; and Su, J. 2025. UniGenCoder: Merging SEQ2SEQ and SEQ2TREE Paradigms for Unified Code Generation. In *2025 IEEE/ACM 47th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, 71–75.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Song, L.; Shi, T.; and Zhao, J. 2025. The hallucination tax of reinforcement finetuning. *arXiv preprint arXiv:2505.13988*.

Sun, Y.; Yin, Z.; Guo, Q.; Wu, J.; Qiu, X.; and Zhao, H. 2024. Benchmarking Hallucination in Large Language Models Based on Unanswerable Math Word Problem. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2178–2188.

Wang, A.; Song, L.; Tian, Y.; Peng, B.; Yu, D.; Mi, H.; Su, J.; and Yu, D. 2025a. LiteSearch: Efficient Tree Search with Dynamic Exploration Budget for Math Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25318–25326.

Wang, W.; Shi, J.; Ling, Z.; Chan, Y.-K.; Wang, C.; Lee, C.; Yuan, Y.; Huang, J.-t.; Jiao, W.; and Lyu, M. R. 2024. Learning to Ask: When LLM Agents Meet Unclear Instruction. *arXiv preprint arXiv:2409.00557*.

Wang, Y.; Li, H.; Zhang, X.; Wu, J.; Liu, X.; Hu, W.; Guo, Z.; Huang, Y.; Xin, Y.; Yang, Y.; Su, J.; Chen, Q.; and Li, S. 2025b. EpiCoder: Encompassing Diversity and Complexity in Code Generation. In *Forty-second International Conference on Machine Learning*.

Wu, S.; Galley, M.; Peng, B.; Cheng, H.; Li, G.; Dou, Y.; Cai, W.; Zou, J.; Leskovec, J.; and Gao, J. 2025. CollabLLM: From Passive Responders to Active Collaborators. In *Forty-second International Conference on Machine Learning*.

Yamauchi, R.; Sonoda, S.; Sannai, A.; and Kumagai, W. 2023. Lpml: llm-prompting markup language for mathematical reasoning. *arXiv preprint arXiv:2309.13078*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Zamfirescu-Pereira, J. D.; Wong, R. Y.; Hartmann, B.; and Yang, Q. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, 1–21.

Zhang, M. J.; Knox, W. B.; and Choi, E. 2025. Modeling Future Conversation Turns to Teach LLMs to Ask Clarifying Questions. In *The Thirteenth International Conference on Learning Representations*.

Zhou, Z.; Song, J.; Yao, K.; Shu, Z.; and Ma, L. 2024. Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2081–2088. IEEE.