Teaching Language Models To Gather Information Proactively

Tenghao Huang[1]* **Sihao Chen**[2] **Muhao Chen**[3]
Jonathan May[1] **Longqi Yang**[2] **Mengting Wan**[2] **Pei Zhou**[2]
[1]University of Southern California, [2]Microsoft, [3]University of California, Davis
tenghaoh@usc.edu, pei.zhou@microsoft.com

Abstract

Large language models (LLMs) are increasingly expected to function as collaborative partners, engaging in back-and-forth dialogue to solve complex, ambiguous problems. However, current LLMs often falter in real-world settings, defaulting to passive responses or narrow clarifications when faced with incomplete or underspecified prompts—falling short of proactively gathering the missing information that is crucial for high-quality solutions. In this work, we introduce a new task paradigm: proactive information gathering, where LLMs must identify gaps in the provided context and strategically elicit implicit user knowledge through targeted questions. To systematically study and train this capability, we design a scalable framework that generates partially specified, real-world tasks, masking key information and simulating authentic ambiguity. Within this setup, our core innovation is a reinforcement finetuning strategy rewards questions that elicit genuinely new, implicit user information—such as hidden domain expertise or fine-grained requirements—that would otherwise remain unspoken. Experiments demonstrate that our trained Qwen-2.5-7B model significantly outperforms o3-mini by **18%** on automatic evaluation metrics. More importantly, human evaluation reveals that clarification questions and final outlines generated by our model are favored by human annotators by **42%** and **28%** respectively. Together, these results highlight the value of proactive clarification in elevating LLMs from passive text generators to genuinely collaborative thought partners.

1 Introduction

Large language models (LLMs) are now indispensable partners for solving diverse reasoning tasks, such as drafting proofs, debugging code, or writing essays, often excelling when provided with well-structured instructions and sufficient information. (Lightman et al., 2023; Chen et al., 2024a; Luo et al., 2024; Wang et al., 2024; Brown et al., 2024; Tian et al., 2024). However, as LLMs become more tightly woven into real-world workflows, the demands on their conversational abilities are evolving. Instead of simply answering queries, users increasingly expect LLMs to act as collaborative partners, engaging in multi-turn, back-and-forth dialogues to tackle complex, ambiguous, and open-ended problems (Bommasani et al., 2022; Spangher et al., 2025a; Handa et al., 2025) (Fig. 1).

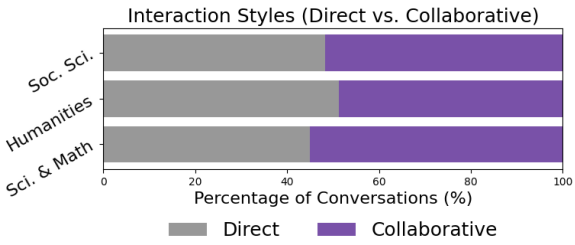

Figure 1: Distribution of LLM Interaction Styles Across Disciplines. Collaborative exchanges account for over half of the interactions in all three fields, indicating a widespread shift from one-shot prompting to iterative, multi-turn dialogue (Handa et al., 2025).

A key challenge in collaborative settings is *information asymmetry*: Users often provide underspecified prompts. Today's models address this by asking *clarification* questions related to ambiguities in the user input, e.g., "Do you want APA or MLA style?". Because these questions are effectively a function of the current input and dialogue prefix alone, they resolve surface uncertainties but rarely change the trajectory toward a substantially better solution. For example, given "Write a policy brief on hospital readmission", asking about length or citation style does little to determine what a strong brief should actually contain.

Motivated by such, we study the task of proactive information gathering. The core research question

*Work done during Tenghao's internship at Microsoft Office of Applied Research.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15588–15599
November 4-9, 2025

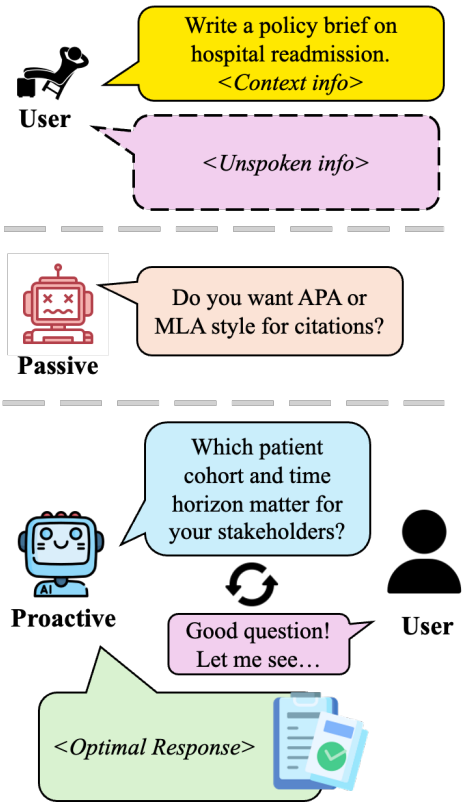

Figure 2: Passive vs. Proactive information gathering. Given an underspecified request, passive models ask surface clarifications (e.g., citation style) that depend only on the observed context. Proactive models ask along new, high-leverage dimensions the user has not supplied (e.g., cohort and time horizon), eliciting constraints that reshape the target solution and enable a higher-utility, *optimal* response.

is *"can language models proactively propose novel ideas, suggestions, or follow-up questions when working interactively with user on a solution?"*

For example, when user asks "Which patient cohort and time horizon matter for your stakeholders?", the model is expected to proactively interact with the user and provide different perspectives to consider in order for a concrete response, e.g. the stakeholder constraints, evaluation criteria, sources to privilege, time horizons, etc. The goal is to shift the LLM from a passive responder to a genuine thought partner that can anticipate user needs, proactively ask for user input, and work interactively with the user to produce more comprehensive responses or artifacts (Fig. 2).

However, designing LLMs that excel at proactive information gathering poses several obstacles. High-quality, collaborative dialogue data is scarce and difficult to scale; organic user logs are often noisy and proprietary, while crowdsourcing nuanced, domain-specific exchanges is expensive and hard to control (Malaviya et al., 2024; Spangher et al., 2025a). Even more fundamentally, the qualities that define a "good" proactive question—helpfulness, novelty, contextual complementarity—are subjective and hard to capture with standard reward signals or simple heuristics.

In this work, we address these challenges by designing a framework that specifically rewards pioneering, context-complementary questions—those that reach beyond what is already provided, proactively soliciting critical information from users. Our approach has three core innovations:

1. **Task Formulation:** We introduce *proactive information gathering* as a new task for LLMs, formalizing the ability to identify and elicit targeted information that would lead to a more comprehensive or in-depth response
2. **Synthetic Conversation Engine:** Leveraging the DOLOMITES dataset (Malaviya et al., 2024), we construct a simulation pipeline that creates ambiguous prompts and rich clarification trajectories by systematically masking critical information—ensuring that proactive questioning becomes necessary for task completion.
3. **Reinforcement Fine-Tuning:** We propose a reward structure that rewards questions reaching beyond the provided context, and fine-tune LLMs using proximal policy optimization. Experiments demonstrate that our trained Qwen-2.5-7B model significantly outperforms o3-mini by **18%** on automatic evaluation metrics. More importantly, human evaluation reveals that clarification questions and final outlines generated by our model are favored by human annotators by **42%** and **28%** respectively.

Together, these contributions chart a path toward LLMs that are not just compliant responders, but proactive, collaborative partners—able to drive richer, more effective conversations by actively seeking the information that matters most. All of the code used in our experiments is publicly available [1].

2 Related Work

Proactive Agent. The term "*proactive agent*" has long been correlated with agents that ask follow-up questions to resolve ambiguities (Bi et al., 2021; Ren et al., 2021). However, these efforts focus on addressing slot ambiguities, which aims to clar-

[1] https://github.com/tenghaohuang/Teaching-Language-Models-To-Gather-Information-Proactively

ify ambiguous information present in corpus (Guo et al., 2021; Deng et al., 2022; Huang et al., 2024; Pang et al., 2024; Chen et al., 2024b). Recent works extend the notion of a 'proactive agent', encompassing agents that *predict* user tasks (Lu et al., 2024). In this work, our definition of proactive agents entails actively identifying incomplete knowledge regarding domain procedures and user preferences through iterative clarification.

Recent works emphasize proactive assistance for everyday conversation (Chen et al., 2024b) within fully unobservable environments and reasoning tasks (Wu et al., 2025). In contrast, our work targets open-ended writing tasks and employs a partially observable environment, reflecting real-world scenarios where users inputs imperfect prompt and agents only have access to partial information up front and must strategically clarify hidden details through iterative dialogue.

Reinforcement Learning for LLM Alignment. Previous works focus on reasoning scenarios where step-level rewards are available or the outcome is easy to verify, such as mathematics (Chen et al., 2024a; Luo et al., 2024; Lightman et al., 2023; Wang et al., 2024), and coding (Brown et al., 2024) tasks. However, these methods are not generalizable to open-ended tasks, where supervision signals are sparse. Our work proposes a framework that masks user domain knowledge and preferences, providing sufficient learnable reward signals for models at train time.

Another line of works uses reward signals from interactive settings (Zhou et al., 2023; Roth et al., 2025), typically relying on simulated user feedback to shape the model's behavior.

3 Task and Dataset

In this section, we formalize the proactive information gathering task (§3.1), describe how it is instantiated using the DOLOMITES dataset (§3.2), and present our evaluation framework (§3.3), which leverages a judge LLM to assess model outputs.

3.1 Task Definition

Let $\mathcal{E}$ denote the **explicit information** provided by the user (e.g., stated goals, facts, and constraints), and let $\mathcal{I}$ represent the **implicit information** (unstated assumptions, domain conventions, and fine-grained requirements) necessary for a complete solution. Let f_θ be a language model parameterized by θ.

The goal is to produce an output $\hat{\mathbf{y}}$ that aligns with the ideal solution $\mathbf{y}^*$, which depends on both explicit and implicit information:

$$\begin{aligned} &\text{Ideal output:} \quad \mathbf{y}^* = f_\theta(\mathcal{E}, \mathcal{I}) \\ &\text{Model output:} \quad \hat{\mathbf{y}} = f_\theta(\mathcal{E}) \end{aligned}$$

The objective is to minimize the alignment gap:

$$\min_\theta \ \mathcal{L}\big(\hat{\mathbf{y}}, \mathbf{y}^*\big)$$

where loss $\mathcal{L}$ measures deviation from the ideal output.

Since $\mathcal{I}$ is not provided, the assistant must proactively infer or elicit the missing information:

$$\hat{\mathbf{y}} = f_\theta(\mathcal{E}, \hat{\mathcal{I}})$$

where $\hat{\mathcal{I}}$ is the assistant's best estimate of the required implicit information, obtained via clarification or reasoning.

3.2 Dataset Adaption

We adapt our data from DOLOMITES datasets (Malaviya et al., 2024). Each instance is a quadruple $\langle \mathbf{o}, \mathbf{p}, \mathbf{i}, \mathbf{s} \rangle$ with the following components:

1. Task **o**bjective ($\mathbf{o}$) — a concise statement of the goal.
2. Task **p**rocedure ($\mathbf{p}$) — domain hints or recommended steps.
3. **I**nput section ($\mathbf{i}$) — facts, constraints, or numerical data.
4. Output **s**pecification ($\mathbf{s}$) — required format, style, and subsections.

The creators elicit 519 task templates from 266 professionals spanning **25 domains** (e.g., medicine, law, civil engineering) and create authentic, real-world writing tasks.

Masking scheme. To align DOLOMITES with our focus on explicit versus implicit information, we adapt each data instance as follows: we first extract the explicit part of each task instance by identifying the task objective ($\mathbf{o}$) and input context ($\mathbf{i}$), which together capture the information a typical user would explicitly provide in a real-world prompt. Formally,

$$\mathcal{E} =< o, i > .$$

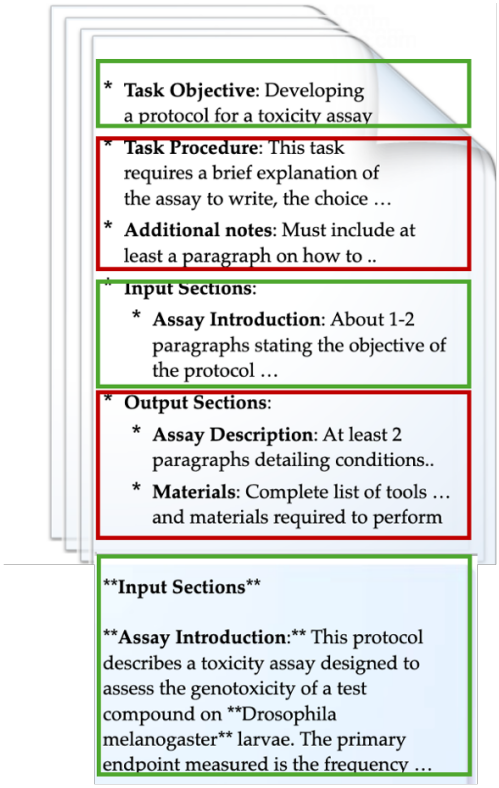

Figure 3: An example of our task input. Contents marked by red boxes are not visible to LLMs. Only contents marked by green boxes are fed to LLMs as input.

In contrast, we define the implicit part as comprising the procedure or domain expertise ($\mathbf{p}$) and the output specification ($\mathbf{s}$). Formally,

$$\mathcal{I} =< p, s > .$$

We mask implicit information that is crucial for a high-quality solution but is rarely stated outright by users. During experiments, we simulate incomplete user prompts by revealing only the explicit information ($\mathcal{E}$) to the model, as shown in Fig. 3. To successfully complete the task, the model must proactively interact with a user oracle—posing clarification questions to uncover the implicit aspects ($\mathcal{I}$) needed-to produce a satisfactory output.

User Response Simulation. To enable controlled, scalable training and evaluation of proactive information gathering, we follow Spangher et al. (2025b) and implement a user conversation simulation engine that mimics realistic interactions between the assistant and a domain expert. When the model poses a question, the simulated user oracle provides answers based strictly on the masked implicit information—ensuring responses are both faithful to the original task author's intent and contextually relevant. We will introduce more about this process in §4.1.

3.3 Evaluation Protocol

For each instance, we let the output specification be distilled into a set of checklist items $\mathbf{s} = \{c_1, \ldots, c_m\}$. An independent, frozen LLM judge evaluates the assistant's response $\hat{\mathbf{y}}$ against each checklist item:

$$\textsc{Match}(c_j, \hat{\mathbf{y}}) \in \{0, 1\},$$

where 1 denotes satisfactory coverage. The overall writing score is computed as:

$$\textsc{Score}(\hat{\mathbf{y}}) = \frac{1}{m} \sum_{j=1}^{m} \textsc{Match}(c_j, \hat{\mathbf{y}}).$$

No credit is given for omitted or incorrectly formatted items, thereby measuring both content and stylistic fidelity. See Fig. 8 for the judge prompt.

4 Method

Our goal is to enable LLMs to *proactively* determine what clarification questions to ask in collaborative problem-solving settings. We frame this as a partially-observable markov decision process (POMDP), and we use reinforcement learning as a way to solve it. A central challenge is the absence of reliable, dense supervision for every proposed question. Instead of rewarding questions themselves, we focus on the outcome—*does the question elicit new information that was not already available to the assistant?* This guiding principle motivates our reward formulation, which rewards the discovery of genuinely missing information.

To systematically study and train such proactive clarification behavior, we develop a synthetic conversation engine that simulates realistic, multi-turn assistant–user dialogues (§4.1). We then detail our reward design (§4.2).

4.1 Synthetic Conversation Engine

Setting. At the start of each episode, the assistant LLM is provided with explicit information $\mathcal{E}$, while the implicit information $\mathcal{I}$ is not availble. A second LLM acting as a **user oracle** has access to both $\mathcal{E}$ and $\mathcal{I}$. Thus, the assistant operates under partial observability.

Dialogue phase. Over up to n turns ($n \leq 5$ in our experiments), the assistant may ask clarification questions, q_t $(t = 1, \ldots, n)$, about the task. The oracle responds with answers $a_t = f_\theta(\mathcal{E}, \mathcal{I})$.

A good question would help uncover consistent implicit information ($\mathcal{I}$). The running dialogue after t turns is $D_t = \{(q_1, a_1), \ldots, (q_t, a_t)\}$.

Draft phase. D_t can be thought of as $\hat{\mathcal{I}}$, which is the assistant's best estimate of the required implicit information After exhausting the turn budget or issuing a `STOP`, the assistant must produce its final output, $\hat{\mathbf{y}} = f_\theta(\hat{\mathcal{I}}, D_t)$.

4.2 Reward Signal Design

Motivation. Rewarding proactive information gathering is non-trivial because each episode involves two intertwined skills: (*i*) formulating a *useful* question q, and (*ii*) evaluating its utility with a reward signal r. A naïve approach would be to evaluate the entire final response $\hat{\mathbf{y}}$, incorporating all user answers. However, this is impractical: long-form outputs (typically 500+ tokens) must be scored against a non-exhaustive reference $\mathbf{y}$, resulting in sparse and often uninformative rewards—especially early in training, when useful questions are rare.

Evidence-sentence reward. We address this by rewarding the *question* based on whether the answer to the question uncovers genuinely missing information. Intuitively, a valuable clarifying question should:

1. Target information *absent* from the explicit information $\mathcal{E}$,
2. Be *answerable* using the implicit information $\mathcal{I}$.

To operationalize this, we perform an *evidence-sentence* check at each dialogue turn t:

Let the implicit information be split into sentences, $\mathcal{I} = \{s_1, \ldots, s_{|\mathcal{I}|}\}$, with indices corresponding to hidden fields $\mathcal{H} = \{1, \ldots, |\mathcal{I}|\}$. Given a question q_t, we prompt the user oracle (LLM) to return the set of sentence indices $A(q_t)$ it would cite when answering q_t:

$$A(q_t) = LLM(q_t, \mathcal{I}) \subseteq \{1, \ldots, |\mathcal{I}|\}.$$

The immediate reward is then:

$$r_t(q_t) = \begin{cases} 1, & \text{if } A(q_t) \cap \mathcal{H} \neq \varnothing, \\ 0, & \text{otherwise.} \end{cases}$$

A question is thus rewarded if it elicits *any* evidence from the hidden fields, incentivizing the model to seek missing information without requiring dense or span-level annotation. We then use this reward signal to train a policy model using Proximal policy optimization (PPO) in an actor–critic framework (Schulman et al., 2017).

5 Experiments

In this section, we evaluate our trained model for proactive information gathering and compare it with baseline methods. We first delve into the details of our experimental setup (§5.1), discuss the results obtained (§5.3), and perform analysis.

5.1 Implementation Details

We fine-tune a Qwen-2.5-7B model for three epochs, using eight A100 GPUs and the verl implementation of PPO. For each training run, we set a turn budget of five steps per episode. Training is performed with a batch size of 256 episodes, using minibatches of 16. The PPO clipping parameter is set to 0.2. We use separate learning rates for the actor and critic—2×10^{-5} for the actor and 1×10^{-4} for the critic. Additionally, we apply gradient norm clipping at 1.0. We fix vanilla GPT-4o as the writer at draft phase throughout our experiments.

5.2 Baselines

To isolate the value of *proactive information gathering* we compare our method against four alternative policies, each instantiated with the same GPT-4o or Qwen-2.5-7B backbone and evaluated under the dialogue budget $n \leq 5$:

GPT-4o Direct. The vanilla GPT-4o that **never** asks clarifying questions. It doesn't seek to resolve uncertainties or fill in missing details by querying the user. It just generates an output directly.

Vanilla LLMs with QA. We employ an in-context prompting strategy [2] that lets the model generate questions to the simulated user in a multi-turn conversation. The conversation will be incorporated before producing the final response. This measures whether a vanilla model's clarification ability can compensate for missing information. Particularly, we include GPT-4o, o3-mini, and Qwen-2.5-7B-Instruct as baseline models.

SFT LLMs on Emulated Conversations. Although raw conversations on proactive clarification between users and LLMs are not available, we synthesize multi-turn conversations between models and users, and perform supervised fine-tuning (SFT) with LLMs. Specifically, we prompt LLM with $\langle \mathbf{o}, \mathbf{p}, \mathbf{i}, \mathbf{s} \rangle$ and ask LLM to generate a synthetic conversation if given $\langle \mathbf{o}, \mathbf{i} \rangle$, how to uncover

[2] Prompt details can be found in Fig. 9

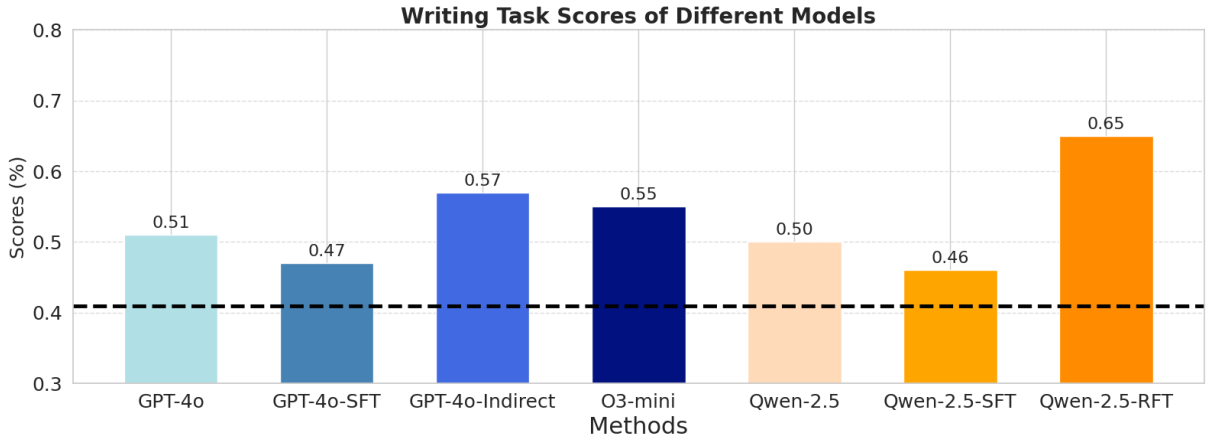

Figure 4: Writing Task Performance Across LLM Variants. Bar chart comparing average writing task scores for several models. Proactively fine-tuned Qwen-2.5-RFT achieves the highest score (0.65), outperforming both base and SFT (supervised fine-tuned) versions of GPT-4o and Qwen-2.5, as well as strong baseline O3-mini. The dashed line marks GPT-4o direct writing performance without asking clarification questions. Our best results (Qwen-2.5-RFT) show statistically significant difference with baselines, $p < 0.05$ (Koehn and Monz, 2006).

$\langle \mathbf{p}, \mathbf{s} \rangle$. We use Azure's AI finetuning service[3] to finetune GPT-4o model and use the `verl` framework (Sheng et al., 2024) to finetune a Qwen-2.5-7B-Instruct model.

GPT-4o Indirectly Supervised. Beyond synthetic data, we realize human brainstorming sessions, though noisy, are also rich in proactive clarification data. QMSUM is a dataset of meeting transcripts in academic, industry and public policy (Zhong et al., 2021). We adapt this dataset, isolating all the conversation turn that proposes questions and train LLM in a DPO fashion (Rafailov et al., 2024).

5.3 Main Results

We summarize the performance of all evaluated models on the writing task in Fig. 4. The main findings are as follows:

Proactive Clarification Substantially Improves Alignment. Our method, Qwen-2.5-RFT, achieves the highest score of 0.65, outperforming all baselines by a large margin. This demonstrates that explicitly training LLMs to proactively ask clarification questions using reinforcement learning leads to significantly better writing task completion, especially in ambiguous or under-specified scenarios.

Supervised Fine-Tuning (SFT) Alone is Insufficient. Both GPT-4o-SFT (0.47) and Qwen-2.5-SFT (0.46) models, which are fine-tuned on synthetic multi-turn conversations, perform worse comparing to their respective base models (GPT-4o: 0.51, Qwen-2.5: 0.50). This indicates that SFT does not robustly endow them with the capacity for contextually proactive questioning in unseen situations, reflecting the challenging nature of our proposed task.

We also observe that GPT-4o-Indirect, trained on real-world brainstorming and meeting data, achieves a competitive score of 0.57, suggesting that exposure to human-driven clarifications provides useful signal. However, it still underperforms compared to our reinforcement-learned model.

5.4 Further Analysis

We conduct a comprehensive quantitative analysis to evaluate the effectiveness of proactive clarification training. The results demonstrate clear and consistent gains from our training piepline for proactive question-asking.

Domain-wise Success Rates. Tab. 1 reports writing task scores across three key domains: social science, technology, and humanities. Our Qwen-2.5-RFT achieves state-of-the-art performance in each domain. We also observe that the performance gains are particularly pronounced in domains such as social science and humanities, where the Qwen-2.5-RFT model outperforms the direct baseline by $+0.37$ and $+0.31$ points, respectively. These domains are typically more open-ended and require deeper contextual reasoning, as opposed to technology, which is often more procedural. The strong gains in these complex subjects indicate the strength of our pipeline for tasks demanding nuanced clarification and richer information gather-

[3] https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/fine-tuning-overview

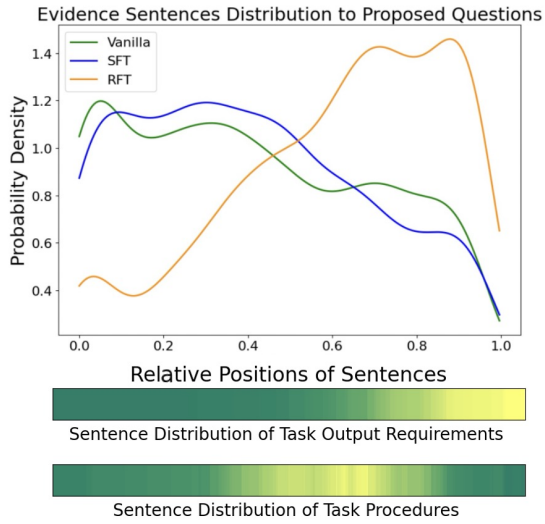

Figure 5: Distribution of Evidence Sentences Supporting Model-Generated Questions. The top plot shows the probability density of sentence normalized positions used as evidence for questions proposed by Qwen-2.5-Instruct vanilla, SFT, and RFT models. The two heatmaps indicate where output requirements and task procedures appear within the source documents. The x-axis represents normalized positions within documents (0.0 = beginning, 1.0 = end). Yellow regions in the heatmaps indicate where certain types of information are most densely concentrated.

ing.

Analysis of Evidence Sentence Distributions. Fig. 5 examines where in the context models locate their evidence when asking clarifying questions. The Qwen-RFT model demonstrates a clear ability to target both the procedural and output-requirement segments of the context. In contrast, the Vanilla and SFT models are more biased toward asking questions on existing information, often failing to pinpoint valuable information in implicit information $\mathcal{I} = \langle \mathbf{p}, \mathbf{s} \rangle$. The bottom heatmaps further confirm that the RFT model's evidence aligns closely with actual distributions of task requirements and procedures, validating its information-seeking behavior.

Impact of Dialogue Length. Fig. 6 reports the impact of question-turn budget on writing performance. While all models benefit from additional clarification rounds, Qwen-RFT exhibits the most pronounced gains, peaking at 5 turns and sustaining high performance with further turns. In contrast, both GPT-4o and GPT-4o-SFT plateau after 3 turns, suggesting diminishing returns without targeted reward optimization.

Qualitative Study. Tab. 2 illustrates sample clarifying questions from different models on the lesson-planning task. We observe that baseline models (GPT-4o-vanilla, SFT, o3-mini) mostly generate generic or surface-level questions, such as asking about learning outcomes or activity details. In contrast, RFT-Qwen consistently produces deeper, more targeted questions—probing for strategies to adapt lessons to student needs and methods to connect prior knowledge to new content. As demonstraed in Tab. 2, these questions directly support the required output sections (Teaching, Practice, Activities) and demonstrate greater pedagogical insight and context awareness.

Method	Domains		
	Soc. Sci.	Technology	Humanity
Direct	0.46	0.41	0.40
GPT with 5-turn QA			
4o_Vanilla	0.56	0.50	0.57
4o_SFT	0.62	0.36	0.45
o3_mini	0.54	0.52	0.52
Qwen with 5-turn QA			
Vanilla	0.65	0.48	0.45
SFT	0.45	0.43	0.50
RFT	**0.83**	**0.53**	**0.71**

Table 1: Writing task scores of each method across three domains.The largest improvements are seen in social science (+0.37) and humanities (+0.31), underscoring the model's robustness and ability to proactively clarify under-specified, open-ended tasks that demand deeper contextual reasoning.

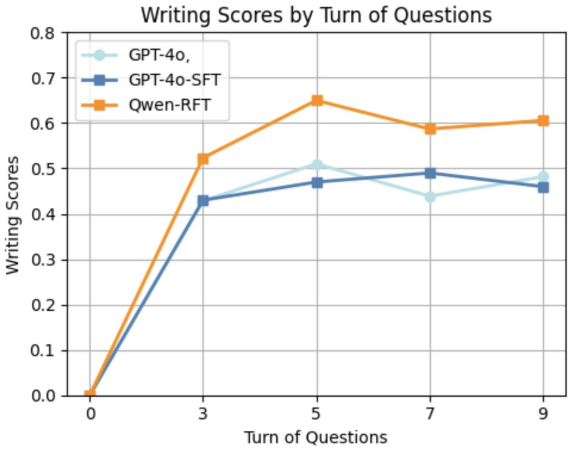

Figure 6: Writing task scores across clarification question Rounds.

Human Evaluation. We further conduct comprehensive human evaluation. Human evaluation reveals that clarification questions and final outlines generated by our model are favored by human annotators by **42%** and **28%** respectively. Together,

Context	Model	Clarifying Question
task_objective: Writing a plan for teaching Maths for Year 2 **National curriculum Year 2** • Solve problems with addition and subtraction using concrete objects and pictorial representations. • Apply mental and written methods. • Recall and use facts to 20 fluently and derive related facts to 100. **National curriculum Year 1** • Read, write, interpret +, –, =. • Add/subtract one- and two-digit numbers to 20. • Solve one-step problems with concrete objects and pictures. **Assessment of previous learning** • Most children add/subtract to 10 confidently. • Some use number bonds; a few still struggle with "equal to".	GPT-4o-vanilla	How will you assess the students' progress throughout the plan to ensure they are meeting the learning outcomes? What specific learning outcomes or goals do you want the students to achieve by the end of the Year 2 Maths plan?
	GPT-4o SFT	Do you have any specific learning outcomes or goals you want to achieve with this plan? What activities or resources will you include in Week 2 to advance their learning?
	o3-mini	Could you share which specific maths topics or concepts you'd like to cover for Year 2? For instance, are you focusing on addition and subtraction, basic geometry, or something else? When planning the transition, how would you guide students from concrete objects to pictorial representations—do you have intermediary steps or bridging activities in mind?
	RFT-Qwen	How can I create a flexible lesson-plan framework that adapts to real-time student engagement and understanding during each class session? How can I structure lesson plans to build on existing knowledge while using varied teaching methods for different learning styles and abilities?

Table 2: Clarifying questions generated by each model for the Year 2 maths lesson-planning task. The prompts probe the three required output sections—*Teaching* (step-by-step teaching plan), *Practice* (pop-up quizzes for in-lesson interactive learning), and *Activities* (reasoning and problem-solving tasks to show mastery for students of different levels). Notably, **RFT-Qwen** consistently asks incisive, on-point questions that align closely with these requirements.

Model (Questions)	Win	Tie	Lose
RFT-Qwen vs o3-mini	62%	18%	20%

Table 3: Comparison of model-generated clarification questions. A "win" indicates that RFT-Qwen's question was preferred over o3-mini's.

Model (Outlines)	Win	Tie	Lose
RFT-Qwen vs o3-mini	50%	28%	22%

Table 4: Comparison of model-generated task outlines. A "win" indicates that RFT-Qwen's outline was preferred over o3-mini's.

these results highlight the value of proactive clarification in elevating LLMs from passive text generators to genuinely collaborative thought partners. We presents detailed analysis in Appx. §B.

6 Conclusion

In this work, we address a fundamental shortcoming of current large language models: their inability to proactively seek out missing, contextually relevant information in ambiguous, open-ended tasks. We formalize proactive clarification as a new benchmark challenge, going beyond traditional clarification and slot-filling to embrace a more collaborative and anticipatory role for LLMs. We present a framework for training LLMs to proactively seek missing information, transforming them from passive responders into active thought partners. By leveraging a synthetic conversation engine and reinforcement fine-tuning, our models consistently outperform strong baselines on both automated metrics and human evaluation, especially in open-ended and under-specified domains. Our results highlight the value of targeted reward optimization for collaborative, context-aware dialogue. We hope this work inspires further research into more realistic multi-turn settings and broader domains, paving the way for LLMs that can engage in richer, more productive human–AI collaboration.

Limitations

While our framework marks a step forward in proactive clarification for LLMs, several limitations remain.

Focus on single benchmark. Our experiments are conducted solely on the DOLOMITES benchmark. However, we argue this does not undermine the generalizability of our method. DOLOMITES is

uniquely comprehensive, spanning 25 professional domains—from humanities and law to technology and medicine—with task instances and evaluation criteria curated by human experts. This diversity ensures our method is evaluated across a wide spectrum of realistic, complex writing scenarios. Moreover, we argue that high-quality benchmarks that combine open-ended writing tasks with fine-grained, expert-driven evaluation criteria remain scarce in the field; DOLOMITES thus provides a strong and meaningful testbed for proactive clarification research.

Future research on multi-turn strategy. Our current approach primarily optimizes for single-turn proactive clarification. While this leads to meaningful improvements in both automated and human evaluations, real-world collaboration often involves more complex, multi-turn interactions—including negotiation, iterative refinement, and the management of evolving user goals. We view the extension of our framework to richer, multi-round conversational settings—possibly incorporating strategies for negotiation and dynamic intent alignment—as an important direction for future work.

Acknowledgement

Muhao Chen was supported by the NSF of the United States Grants ITE 2333736 and OAC 2531126, and the DARPA FoundSci Grant HR00112490370. This work was partially funded by the Defense Advanced Research Projects Agency with award HR00112220046.

References

Keping Bi, Qingyao Ai, and W. Bruce Croft. 2021. Asking clarifying questions based on negative feedback in conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 157–166, New York, NY, USA. Association for Computing Machinery.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process. *Preprint*, arXiv:2405.03553.

Maximillian Chen, Ruoxi Sun, Sercan Ö Arık, and Tomas Pfister. 2024b. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. *arXiv preprint arXiv:2406.00222*.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. Pacific: towards proactive conversational question answering over tabular and textual data in finance. *arXiv preprint arXiv:2210.08817*.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-coqa: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.

Kunal Handa, Drew Bent, Alex Tamkin, Miles McCain, Esin Durmus, Michael Stern, Mike Schiraldi, Saffron Huang, Stuart Ritchie, Steven Syverud, Kamya Jagadish, Margaret Vo, Matt Bell, and Deep Ganguli. 2025. Anthropic education report: How university students use claude.

Tenghao Huang, Dongwon Jung, Vaibhav Kumar, Mohammad Kachuee, Xiang Li, Puyang Xu, and Muhao Chen. 2024. Planning and editing what you retrieve for enhanced tool learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 975–988, Mexico City, Mexico. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the workshop on statistical machine translation*, pages 102–121. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, and 1 others. 2024. Proactive agent: Shifting llm agents from reactive responses to active assistance. *arXiv preprint arXiv:2410.12361*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

Chaitanya Malaviya, Priyanka Agrawal, Kuzman Ganchev, Pranesh Srinivasan, Fantine Huot, Jonathan Berant, Mark Yatskar, Dipanjan Das, Mirella Lapata, and Chris Alberti. 2024. Dolomites: Domain-specific long-form methodical tasks. *Preprint*, arXiv:2405.05938.

Jing-Cheng Pang, Heng-Bo Fan, Pengyuan Wang, Jia-Hao Xiao, Nan Tang, Si-Hang Yang, Chengxing Jia, Sheng-Jun Huang, and Yang Yu. 2024. Empowering language models with active inquiry for deeper understanding. *arXiv preprint arXiv:2402.03719*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to ask appropriate questions in conversational recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 808–817, New York, NY, USA. Association for Computing Machinery.

Nicholas Roth, Christopher Hidey, Lucas Spangher, William F Arnold, Chang Ye, Nick Masiewicki, Jinoo Baek, Peter Grabowski, and Eugene Ie. 2025. Factored agents: Decoupling in-context learning and memorization for robust tool use. *arXiv preprint arXiv:2503.22931*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

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*.

Alexander Spangher, Tenghao Huang, Philippe Laban, and Nanyun Peng. 2025a. Creative planning with language models: Practice, evaluation and applications. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pages 1–9, Albuquerque, New Mexico. Association for Computational Linguistics.

Alexander Spangher, Michael Lu, Sriya Kalyan, Hyundong Justin Cho, Tenghao Huang, Weiyan Shi, and Jonathan May. 2025b. NewsInterview: a dataset and a playground to evaluate LLMs' grounding gap via informational interviews. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32895–32925, Vienna, Austria. Association for Computational Linguistics.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *Preprint*, arXiv:2312.08935.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025. Collabllm: From passive responders to active collaborators. *Preprint*, arXiv:2502.00640.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and 1 others. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.

Pei Zhou, Andrew Zhu, Jennifer Hu, Jay Pujara, Xiang Ren, Chris Callison-Burch, Yejin Choi, and Prithviraj Ammanabrolu. 2023. I cast detect thoughts: Learning to converse and guide with intents and theory-of-mind in dungeons and dragons. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11136–11155.

A Training Results

Validation Reward. Figure 4 illustrates the learning dynamics of our PPO critic over roughly 70 training steps. The average return remains near zero for the first ten steps—corresponding to a

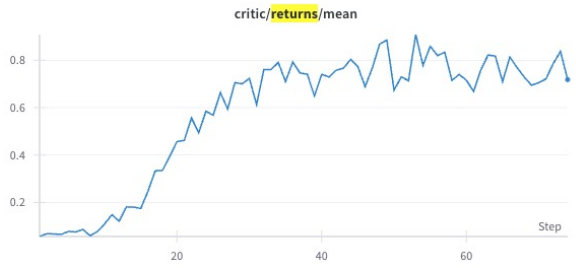

Figure 7: Critic return rewards average per step.

cold-start phase in which the policy is still exploring—then rises sharply between steps 10 and 35 as the model begins to exploit informative questions and receive consistent positive feedback. After reaching the 0.7–0.8 range, the curve displays a saw-tooth pattern: returns fluctuate around a high mean, with intermittent peaks above 0.8 that reflect successful episodes.

B Human Evaluation

To assess the practical utility of the model outputs, we recruited three annotators with graduate-level backgrounds. Annotators were presented with 30 side-by-side outputs from RFT-Qwen and o3-mini for the same input and asked to indicate which model's output they preferred for question generation and final drafted response. Since full written essay responses are more than 500 words, we instruct the model to generate actionable outlines of response instead that are easier for annotators to evaluate. We perform manual inspection and confirm evaluation of full essay responses and corresponding outlines are equivalent.

We also realize there are questions not directly related to implicit information $\mathcal{I}$ also make sense but not evaluated properly, so we recruit humans to evaluate the question quality by itself. Particularly, for question evaluation, our focus is on brainstorming quality: annotators specifically judged each question for how *helpful* it would be in a brainstorming session. In particular, a good brainstorming question should be *inspiring*—that is, it should encourage creative thinking, open up new perspectives, and invite the user to explore ideas beyond the immediate context. Annotators were instructed to prefer questions that spark further thought, rather than questions that simply clarify surface details.

For outline evaluation, annotators considered clarity, completeness, and how well the outline meets the task output requirements, which are initially hidden to the models. The results in Tab. 3 and Tab. 4 show that RFT-Qwen consistently outperforms o3-mini in both question generation and outline production according to human annotators. Notably, our experiments use **Qwen-2.5-7B** as the base model for RFT-Qwen, and yet it outperforms o3-mini, a model recognized for its strong reasoning capabilities. This demonstrates that RFT-Qwen's approach not only produces more human-preferred outputs, but does so even when compared to a larger and well-established reasoning model.

C Prompt Details

In this section, we showcase prompt details. Fig. 8 shows the LLM-as-judge prompt for evaluation. Fig. 9 showcases our prompt strategy for instructing models to generate proactive clarification questions.

Grade the step-by-step outline based on the checklist.
If the step-by-step outline meets the criteria, then the grade +1.
If the step-by-step outline does not meet the criteria, then the grade +0.

The output should be in json format:

```
{
    "grade": "integer",
    "out_of": "integer"
# the total number of criteria in the checklist,
    "comments": "comments",
}
```

Figure 8: Prompt details for LLM-as-Judge evaluation.

You are an AI assistant helping a user with a writing task. Guide this brainstorming session **proactively** and clearly, following these guidelines:

1. **Task Context:** Help the user complete their writing task (content, editing, or brainstorming). Aim for clear, productive results.
2. **Proactive Clarification:** If the task or their instructions are ambiguous, ask clarifying questions only if it helps with the writing. A good question should be creative and spark user thoughts.
3. **Stay on Task:**
   - Respect "out of scope" topics. If the user says a topic is off-limits, don't push—pivot to something relevant.
   - If more information is needed, ask focused follow-up questions.
4. **Tips**
   - Do not focus on clarifying existing details.

Input Task Objective:
<Input Task Objective>

Input Sections:
<Input Sections>

Given the above input, please ask your questions.

Figure 9: Prompt details for proactive clarification question generation.