

CA-BED: Conversation-Aware Bayesian Experimental Design

Anonymous submission

Abstract

Large Language Models (LLMs) excel at static reasoning tasks, yet their reliability diminishes in interactive scenarios where information must be actively acquired through questioning. A key challenge lies in asking questions that minimize uncertainty while simultaneously internalizing responses that may be ambiguous or partially specified. To address this, we propose Conversation-Aware Bayesian Experimental Design (CA-BED), an inference-time probabilistic dialogue planning framework that integrates Bayesian Experimental Design with LLM-based likelihood estimation to optimize question selection over multiple conversational turns. CA-BED models uncertainty continuously, anticipates potential answers, and propagates expected information gain through a simulated conversation tree, enabling more efficient and robust information acquisition. Across entity-deduction benchmarks, CA-BED yields an average 25% improvement in success rates over direct prompting, with comparable gains relative to alternative information-seeking methods. It further achieves a 10.25% average reduction in conversation length, indicating greater efficiency in information gathering. These results highlight CA-BED’s effectiveness as a principled framework for reliable interactive reasoning in real-world settings.

1 Introduction

Large Language Models (LLMs) excel at static reasoning tasks such as code generation and logic puzzles (Huang and Yang 2025; Xu et al. 2025b; Wang et al. 2025b; Jiang et al. 2024), yet their performance declines in interactive settings requiring active information gathering. Persistent weaknesses appear in entity deduction (Zhang, Lu, and Jaitly 2024; Zhou et al. 2025; Bertolazzi et al. 2023), tool use (Patil et al. 2025), and clarification tasks (Gan et al. 2024). In domains like tutoring, diagnosis, and troubleshooting, effective questioning is essential; models must decide not only how to reason, but when and what to ask. Because exhaustively defining all possible queries is infeasible, adaptive information acquisition remains a key challenge for reliable LLM deployment.

Recent approaches address this through (1) fine-tuning models to generate questions (Wu et al. 2025; Zhang, Knox, and Choi 2025; Andukuri et al. 2024; Li et al. 2025), which often struggle to generalize, and (2) training-free planning methods where models simulate dialogue trajectories before

acting (Hu et al. 2024; Yu, Chen, and Yu 2023; Li et al. 2024). The Uncertainty of Thoughts (UoT) framework (Hu et al. 2024) treats conversation as a decision process over a contextual tree, selecting questions that partition the hypothesis space efficiently.

However, UoT and similar methods (Chan et al. 2025; Chopra and Shah 2025) falter with partial or ambiguous evidence. Real-world queries like “Do you have a fever?” should adjust, rather than eliminate, hypotheses. Allowing soft partitions mitigates this but often extends dialogue unnecessarily. A principled approach is needed to represent graded evidence and propagate uncertainty through the conversation.

We propose **Conversation-Aware Bayesian Experimental Design (CA-BED)**, which integrates Bayesian Experimental Design (BED) with conversational planning to enhance efficiency and robustness in LLM-based information seeking. BED optimizes data acquisition by maximizing expected information gain (Denzler and Brown 2002; Berry 2006; Choudhury et al. 2025); CA-BED extends this to dialogue, treating each question as a Bayesian experiment that updates beliefs over hypotheses.

Contributions:

- **Uncertainty-aware conversation planning:** A hybrid framework combining UoT with probabilistic belief modeling and LLM-based likelihood estimation for dynamic uncertainty reasoning.
- **Multi-answer planning:** Anticipating diverse responses improves information efficiency and robustness, capturing the ambiguity of real dialogue.
- **Empirical validation:** CA-BED outperforms prior baselines on *Detective Cases* and *20 Questions*, improving both accuracy and conversational efficiency.

2 Related Work

Research on LLM reasoning has focused primarily on static tasks like math and logic (Wang et al. 2025b; Lin et al. 2025; Xu et al. 2025a), while interactive reasoning, where models actively seek information, remains underexplored. Prior work distinguishes clarification tasks, which resolve ambiguous prompts (Tanjim et al. 2025; Aliannejadi et al. 2019; Deng et al. 2023), from information-seeking tasks,

which study how LLMs query or use tools to obtain missing facts (Handa et al. 2024; Patil et al. 2025).

Information-seeking benchmarks often use entity-deduction games like Twenty Questions (Bertolazzi et al. 2023; Zhang, Lu, and Jaitly 2024), Guess the Celebrity (Choudhury et al. 2025), and medical diagnosis (Li et al. 2024). Larger setups such as AR-bench (Zhou et al. 2025) extend this with multiple speaking agents and show that LLMs still lag behind humans.

Some works fine-tune models to ask better questions (Wu et al. 2025; Zhang, Knox, and Choi 2025), while others use online reasoning guided by heuristics like information gain or entropy (Chan et al. 2025; Choudhury et al. 2025; Hu et al. 2024; Wang et al. 2025a; Cooper et al. 2025). Methods such as UoT (Hu et al. 2024) perform well but struggle in large or uncertain search spaces (Zhou et al. 2025).

Many of these approaches can be seen as special cases of Bayesian Experimental Design (BED), which selects actions that maximize expected information gain (Denzler and Brown 2002; Berry 2006; Rainforth et al. 2023). BED-LLM (Choudhury et al. 2025) applies this idea by evaluating candidate questions, but its strategy is short-sighted, optimizing only the next query rather than long-term conversational efficiency.

3 Methodology

We propose the Conversation-Aware Bayesian Experimental Design (CA-BED) algorithm (Figure 2), which extends the Uncertainty of Thoughts (UoT) framework (Hu et al. 2024) and builds on recent advances such as BED-LLM (Choudhury et al. 2025).

3.1 Probabilistic Model

Hypothesis Space. Let $\mathcal{H} = \{h_1, \dots, h_n\}$ denote a discrete set of mutually exclusive hypotheses.

Belief State. At step k , the belief distribution over \mathcal{H} given observed evidence $E_{1:k}$ is

$$B_k(h_i) = P(h_i | E_{1:k}). \quad (1)$$

The prior $B_0(h_i) = P(h_i)$ is typically uniform, though informative priors may also be used.

3.2 Inference

Evidence. Each user response E_k obtained from question Q_k constitutes new evidence.

Likelihood Estimation. An LLM estimates (logprobs) the likelihood $P(E_k | h_i)$ for each hypothesis h_i . To account for epistemic uncertainty, we apply a confidence-weighted smoothing:

$$P'(E_k | h_i) = \epsilon P(E_k | h_i) + (1 - \epsilon)u, \quad (2)$$

where ϵ is a confidence coefficient and u a uniform baseline. For simplicity, we drop the prime notation in subsequent equations.

Belief Update. Assuming conditional independence of evidence given h_i , the belief state is updated via Bayes' rule:

$$P(h_i | E_{1:k}) = \frac{P(h_i | E_{1:(k-1)})P(E_k | h_i)}{\sum_j P(h_j | E_{1:(k-1)})P(E_k | h_j)}. \quad (3)$$

3.3 Conversation Tree

The dialogue is represented as a tree alternating between *question* and *evidence* nodes:

- **Evidence nodes:** belief states B_k after observing evidence E_k .
- **Question nodes:** candidate questions Q_k , each branching by possible user responses.

3.4 Optimal Question Selection

At each evidence node, the next question Q is chosen to maximize expected information gain.

Entropy and Information Gain. Belief uncertainty is quantified using Shannon entropy:

$$H(B) = - \sum_i P(h_i) \log_2 P(h_i), \quad (4)$$

and the expected entropy reduction defines the information gain:

$$IG(Q) = H(B_{\text{parent}}) - \mathbb{E}_{E_j \sim P(E_j)}[H(B_j)]. \quad (5)$$

Reward Functions. To discourage unbalanced branching, the immediate reward is defined as

$$R_I(Q) = \frac{IG(Q)}{1 + \lambda(\max_j P(E_j) - \min_j P(E_j))}, \quad (6)$$

where $\lambda \geq 0$ regularizes questions whose response probabilities are highly uneven.

To propagate rewards through the conversation tree, we define two recursive quantities:

Accumulated reward. The accumulated reward $R_A(Q)$ at question node Q is the sum of its immediate reward and that of its parent:

$$R_A(Q) = R_I(Q) + R_A(\text{parent}(Q)). \quad (7)$$

Expected reward. The expected reward $R_E(Q)$ represents the total expected utility of asking Q :

$$R_E(Q) = \begin{cases} R_A(Q), & \text{if } Q \text{ is a leaf node,} \\ \mathbb{E}_{E \sim P(E)}[\overline{R_E}(E)], & \text{otherwise.} \end{cases} \quad (8)$$

Here, $\overline{R_E}(E)$ is the average expected reward of the child question nodes that follow from evidence E :

$$\overline{R_E}(E) = \frac{1}{\rho_E} \sum_{z=1}^{\rho_E} R_E(Q'_{(E,z)}), \quad (9)$$

where ρ_E is the number of child question nodes under evidence node E .

3.5 Algorithm Summary

CA-BED selects the next query via lookahead simulation at each evidence node.

Hyperparameters. M_q (candidate questions), D_{sim} (lookahead depth), D_{ask} (conversation limit), ϵ (LLM likelihood confidence), and $\tau_{\text{confidence}}$ (confirmation threshold).

Procedure.

1. *Generate*: Prompt the LLM for up to M_q candidate questions.
2. *Simulate*: Expand each question into a simulation tree up to depth D_{sim} .
3. *Evaluate*: Compute accumulated rewards (R_A) at the leaf nodes.
4. *Backpropagate*: Estimate expected rewards (R_E) from leaves to roots.
5. *Select*: Choose the question with the highest R_E .
6. *Execute*: Ask the selected question, update the evidence node, and repeat.
7. *Terminate*: Stop when belief in any hypothesis h_i exceeds $\tau_{\text{confidence}}$ or depth D_{ask} .

4 Experimental Setup

4.1 Research Questions

We aim to evaluate the effectiveness of CA-BED for improving information acquisition in conversational reasoning. Specifically, our experiments address the following questions:

- **RQ1**: Does CA-BED increase information acquisition over zero-shot prompting?
- **RQ2**: Do explicit likelihoods offer better information acquisition than UoT?
- **RQ3**: Can answer planning improve information acquisition?
- **RQ4**: Does CA-BED increase information acquisition even in unsuccessful tasks?

4.2 Tasks and Benchmarks

We evaluate the above questions on two practical entity-deduction scenarios:

- **20 Questions**: A classic guessing game in which one player asks up to twenty questions to identify a hidden entity selected by another player. We use the COMMON dataset from UoT (Hu et al. 2024), consisting of 111 animals, objects, places, and foods.
- **Detective Cases**: A murder mystery game introduced by (Zhou et al. 2025) of 100 different scenarios. An agent is provided with a scenario and five suspects, and can interrogate each suspect to deduce the murderer.

4.3 Compared Methods

To analyze CA-BED’s contributions, we compare four configurations, each using Deepseek V3.2-Chat as the **Questioner** and Deepseek V3.2-Reasoner as the **Answerer**:

- **Direct Prompting**: The Questioner LLM receives the conversation history and generates either another question or a final prediction.
- **Uncertainty of Thoughts (UoT)**: A special case of CA-BED reflecting prior baselines (Hu et al. 2024) where $\epsilon = 1.0$ and $\tau_{\text{confidence}} = 1.0$. Each answer has a binary likelihood (1 or 0). We set the branching width to 3 and the lookahead depth to 2.

- **CA-BED**: Our proposed method, using a branching width of 3, a lookahead depth of 2, $\epsilon = 0.7$, and $\tau_{\text{confidence}} = 0.8$. The Questioner is restricted to binary (“yes”/“no”) questions, with logprob likelihoods sampled from the Questioner LLM.
- **CA-BED + Answer Planning**: An extension of CA-BED that allows the Questioner to generate and evaluate multiple candidate answers using the same hyperparameters.

5 Results & Discussion

5.1 RQ1: Effectiveness of Conversation-Aware Bayesian Experimental Design

Table 1: Results on *Detective Cases* and *20 Questions*. SR = success rate (%); MC = mean conversation length; MSC = mean length in successful cases.

Method	Detective Cases			20 Questions		
	SR↑	MC↓	MSC↓	SR↑	MC↓	MSC↓
Direct Prompting	18	19.02	17.39	53	15.86	13.81
UoT	24	2.59	2.67	41	6.93	7.07
CA-BED	33	17.20	16.70	86	12.73	11.71
CA-BED + Answer-Planning	25	11.50	10.04	88	12.52	12.34

As shown in Table 1, CA-BED substantially outperforms direct prompting across both benchmarks. In *Detective Cases*, it nearly doubles the success rate (18% → 33%) while using 10% fewer questions. Among successful runs, dialogue lengths are comparable, indicating that CA-BED is more informative within similar interaction spans. On *20 Questions*, it maintains this advantage, improving accuracy by 33% while preserving efficiency despite the task’s shorter dialogues.

Compared to UoT, CA-BED yields a 9% absolute gain on *Detective Cases* but with longer conversations (6.6×). This reflects UoT’s strategy of aggressively halving the hypothesis space, which converges quickly but falters under ambiguous evidence. A similar pattern emerges in *20 Questions*, where CA-BED attains a 45% higher success rate while roughly doubling dialogue length. These results underscore CA-BED’s strength in probabilistic belief updates, which modulate hypothesis probabilities rather than performing hard eliminations, as illustrated in Figure 1.

5.2 RQ2: Do Explicit Likelihoods Improve Information Acquisition over UoT?

To investigate whether the performance gap between UoT and CA-BED can be reduced by incorporating estimator confidence, we extended UoT with a probability-based belief state, but maintained the aggressive splitting of the hypothesis set but with an estimator confidence. So, if a hypothesis is categorised into a “Yes” answer, instead of evaluating on a $P(\text{Yes}|h) = 1.0$, it will be $P(\text{Yes}|h) = 1.0 \cdot 0.7 + 0.5 \cdot 0.3 = 0.85$. We evaluated this change using an estimator confidence of $\epsilon = 0.7$, and further varied the

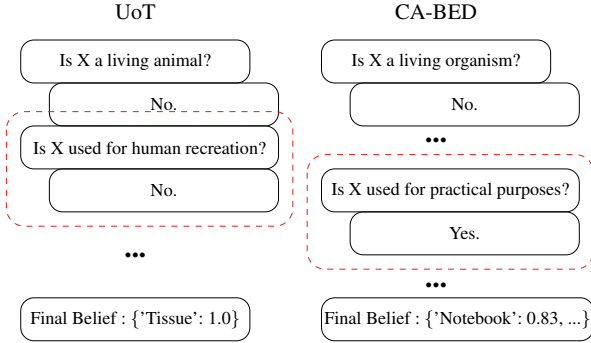


Figure 1: In the face of an ambiguous question UoT prematurely prunes the correct answer 'Notebook' from its belief state which completely derails its conversation trajectory

Table 2: Comparison of CA-BED and UoT augmented with epistemic uncertainty. Values in parentheses indicate $(\epsilon, \tau_{\text{confidence}})$, representing the estimator confidence and confidence threshold parameters, respectively

Method	Detective Cases			20 Questions		
	SR \uparrow	MC \downarrow	MSC \downarrow	SR \uparrow	MC \downarrow	MSC \downarrow
UoT (1.0, 1.0)	24	2.59	2.67	41	6.93	7.07
UoT (0.7, 0.8)	18	7.11	7.44	76	15.02	13.81
UoT (0.7, 1.0)	15	19.11	17.27	65	17.00	18.18
CA-BED (0.7, 0.8)	33	17.20	16.70	86	12.73	11.71

confidence threshold, $\tau_{\text{confidence}} \in \{0.8, 1.0\}$, in case there are inherent limits in confidence between each strategy.

On *20 Questions*, augmenting UoT with estimator confidence ($\epsilon = 0.7$) and a lower threshold ($\tau_{\text{confidence}} = 0.8$) yields a large success rate (SR) gain over baseline (41% \rightarrow 76%). This is most likely because the correct answer is being pruned less quickly. Conversations are also longer, where mean conversation length (MC) increases to 15.02 turns, about 18% higher than CA-BED's 12.73 turns. Raising the threshold to $\tau_{\text{confidence}} = 1.0$ (with $\epsilon = 0.7$ fixed) lowers SR to 65%, suggesting that forcing near certainty is counterproductive for UoT's aggressive splitting strategy.

In *Detective Cases*, there is no improvement, where adding estimator confidence reduces SR from the 24% baseline to 18% at $\tau_{\text{confidence}} = 0.8$ and to 15% at $\tau_{\text{confidence}} = 1.0$. This likely reflects the very small hypothesis set (five suspects), where aggressive splitting remains the bottleneck.

5.3 RQ3: Can Answer Planning Improve Information Acquisition?

In realistic dialogue, answerers are rarely limited to binary *Yes/No* responses. Modeling continuous or nuanced outputs, however, often renders exact Bayesian updates intractable. To address this, we extend CA-BED with an answer-planning mechanism that generates multiple candidate answers and selects the one semantically closest to the answerer's response using Sentence-BERT (Reimers and Gurevych 2019).

As shown in Table 1, this extension yields an average 2% improvement in success rate with comparable dialogue lengths. However, performance decreases by 8% on the *Detective Cases* task. Qualitative analysis suggests that this degradation arises from incomplete coverage of candidate answers, causing semantically mismatched selections and, consequently, inaccurate likelihood estimates.

5.4 RQ4: Does CA-BED Increase Information Gain in Unsuccessful Tasks?

Table 3: Relationship between conversational informativeness and downstream task success. *Top3* (%) = proportion where the correct answer appears in the top-3 candidates. *Post-Dialogue SR* (%) (PD-SR) = accuracy of an external reasoner (DeepSeek V3.2-Reasoner) given each conversation transcript.

Method	Detective Cases		20 Questions	
	Top3 \uparrow	PD-SR \uparrow	Top3 \uparrow	PD-SR \uparrow
Direct	65	24	60	80
UoT	24	24	41	44
CA-BED	66	34	90	93

Prior work on information-seeking strategies (Hu et al. 2024; Chopra and Shah 2025; Choudhury et al. 2025) often uses overall success rate as a proxy for information acquisition, overlooking cases where a model gathers useful evidence without producing the correct final answer. To better isolate this relationship, we re-evaluate CA-BED, UoT, and Direct Prompting using two complementary metrics: (i) Top-3 accuracy, via an oracle that indicates prediction correctness, and (ii) post-dialogue success, measured by giving each conversation to an external reasoner (DeepSeek V3.2-Reasoner) for final prediction.

As shown in Table 3, CA-BED improves Top-3 accuracy by 30% over Direct Prompting and 49% over UoT on 20 Questions. Consistent gains can be seen in *Detective Cases*, though less apparent due to the small size (5) of the hypothesis set. Post-dialogue evaluation further shows 10% and 13% higher success for CA-BED on the two benchmarks, indicating that its conversations convey more informative and reusable evidence than competing methods.

6 Conclusion

We presented Conversation-Aware Bayesian Experimental Design (CA-BED), a probabilistic framework for inference-time dialogue planning. CA-BED unites Bayesian Experimental Design with LLM-based likelihood estimation to optimize question selection across multiple conversational turns. By modeling uncertainty and anticipating possible answers, it achieves more robust and efficient information gathering. On entity-deduction benchmarks, CA-BED improved success rates by 25% over direct prompting and surpassed planning methods such as UoT, while reducing conversation length by 10.25%. These results demonstrate that probabilistic reasoning over uncertainty offers a principled path toward more reliable interactive reasoning in LLMs.

A Prompts

A.1 20 Question Prompts

Question Generation Prompt (20 Questions) [UoT, CA-BED]

You are an expert player of the 20 Questions game. Your goal is to guess a secret entity, X. I will be impersonating the secret entity, X. You will ask me up to 20 questions which start with 'Is X' and can only be answered by 'Yes' or 'No', and I will answer each one truthfully based on being X.

The game has proceeded as follows:
{qa_history}

The secret object is most likely one of the following items:
{belief_state}

Your task is to generate {max_question_nodes} *excellent* yes/no questions to ask next. The best questions are those that will help distinguish between these likely possibilities. Format your response in this structure:

1. <Question 1>
2. <Question 2>
- ...
- n. <Question n>

Question Generation Prompt (20 Questions) [CA-BED + Answer-Planning]

You are an expert player of the 20 Questions game. Your goal is to guess a secret entity, X. I will be impersonating the secret entity, X. You will ask me up to 20 questions which start with 'Is X' and can only be answered by 'Yes' or 'No', and I will answer each one truthfully based on being X.

The game has proceeded as follows:
{qa_history}

The secret object is most likely one of the following items:
{belief_state}

Your task is to generate {max_question_nodes} *excellent* yes/no questions to ask next. The best questions are those that will help distinguish between these likely possibilities. Format your response in this structure:

1. <Question 1>
2. <Question 2>
- ...
- n. <Question n>

Question Generation Prompt (20 Questions) [CA-BED + Answer-Planning]

You are an expert player of the 20 Questions game. Your goal is to guess a secret entity, X. I will be impersonating X.

You may ultimately ask up to {max_conversation_depth} questions during the full game, but right now your task is to propose the next {max_question_nodes} candidate questions.

The secret entity is guaranteed to be one of the following possibilities:
{remaining_hypotheses}

The game so far:
{history_formatted}

Task

Generate questions that help differentiate the remaining candidates. Each question must:

- Start with EXACTLY 'Is X' and be answerable truthfully about the secret entity.
- Provide a realistic set of possible answers for that question.
- Focus on attributes that meaningfully split the hypothesis space.

Response Format

Respond using one line per question in this EXACT structure:

1. Is X ... ?|Answer1|Answer2|Answer3
2. Is X ... ?|Answer1|Answer2
- ...
- n. Is X ... ?|Answer1|Answer2|Answer3|Answer4

Example

1. Is X a living being?|Yes|No
2. Is X commonly found indoors?|Yes|No|Sometimes

Likelihood Prompt (20 Questions) [UoT, CA-BED]

You are playing a game of 20 Questions.

Conditional Assumption

For the purpose of this question, assume {item} is the secret entity.

Scenario
You asked the following question:
"{question}"

Possible Answers
{answer_list_str}

Task
Given that {item} is the secret entity, which answer did the answerer give?
Respond with the number for the answer only.

The answer was number:

Answer Prompt (20 Questions) [UoT, CA-BED]

You are a player of the 20 Questions game.
Your goal is to impersonate the secret entity, X.
Secret Entity
X is {task_answer}.
I will ask up to 20 questions and you should answer each one truthfully based on being X.

Instructions
- Answer truthfully based on what X is.
- You must ONLY respond with either **Yes** or **No**, matching it EXACTLY.
- Do not add extra text or commentary. Return exactly one of the options.

Question
"{current_question}"

A.2 Detective Cases Prompts

Question Generation Prompt (Detective Cases) [UoT, CA-BED]

You are a detective investigating a murder. You can ask up to {max_conversation_depth} questions.

Case Background
{background_info}

The investigation focuses on 5 suspects:
{suspects_info}

{conversation_history_section}

Task

Generate {max_question_nodes} excellent yes/no interrogation questions. - Each question must be explicitly directed to a specific suspect. - Format each question as: "[Suspect Name] Question text". - Each question must be answerable strictly with "Yes" or "No". - Focus on questions that help distinguish between suspects (e.g., motive, alibi, opportunity, or access to the weapon).

Response Format
One line per question:
1. <Question 1>
2. <Question 2>
...
n. <Question n>

Example
1. [Mr. Jones] Were you outside at 12:00 PM?
2. [Dr. Otto] Did you have access to the murder weapon?

Question Generation Prompt (Detective Cases) [CA-BED + Answer-Planning]

You are a detective investigating a murder. You can ask up to {max_conversation_depth} questions.

Case Background
{background_info}

The investigation focuses on 5 suspects:
{suspects_info}

{conversation_history_section}

Task
Generate {max_question_nodes} excellent interrogation questions.
- Each question must be explicitly directed to a specific suspect.
- Format the question as: "[Suspect Name] Question text".
- Provide a realistic set of possible answers for that suspect.
- Focus on questions that help distinguish between suspects (motive, alibi, opportunity, access to weapon).

Response Format
One line per question:
1. <Question 1>|Answer1|Answer2|Answer3
2. <Question 2>|Answer1|Answer2
...
n. <Question n>|Answer1|Answer2|...|AnswerK

Example

1. [Mr. Jones] Where were you at the time of the murder?|In the kitchen|In the garden|With the victim
2. [Dr. Otto] Did you have access to the murder weapon?|Yes|No

Answer Generation Prompt (Detective Cases) [UoT, CA-BED, Direct]

You are roleplaying as a suspect in a murder investigation.

Suspect

- Name: {suspect.name}
- Task: {suspect.task}
- Story: {suspect.story}

Instructions

- Answer the detective's question in character as {suspect.name}.
- Stay consistent with your task and story.
- You may lie, evade, or tell the truth depending on what seems natural for this suspect.
- You must ONLY respond with either **Yes** or **No**, matching it EXACTLY.
- Do not add extra text or commentary. Return exactly one of the options.

Detective's Question
"{question}"

Answer Generation Prompt (Detective Cases) [CA-BED + Answer-Planning]

You are roleplaying as a suspect in a murder investigation.

Suspect

- Name: {suspect.name}
- Task: {suspect.task}
- Story: {suspect.story}

Instructions

- Answer the detective's question in character as {suspect.name}.
- Stay consistent with your task and story.
- You may lie, evade, or tell the truth depending on what seems natural for this suspect.
- You must ONLY respond with one of the following option, matching it EXACTLY: {answers}

- Do not add extra text or commentary. Return exactly one of the options.

Detective's Question
"{question}"

Likelihood Prompt (Detective Cases) [UoT, CA-BED, CA-BED + Answer-Planning]

You are a detective investigating a murder case.

Case Background
{background_info}

Suspects
{suspects_info}

Conditional Assumption

For the purpose of this question, assume {hypothesis_name} is the murderer.

Scenario

You asked {answerer_name} the following question:
"{actual_question}"

Possible Answers
{answer_list_str}

Task

Given that {hypothesis_name} is the murderer, which answer did {answerer_name} give?
Respond with the number for the answer only.

{answerer_name}'s answer was number:

Questioner Prompt (Detective Cases) [Direct]

You are a detective investigating a murder.

Case Background
{case_background}

The investigation focuses on {num_suspects}

suspects:
{suspects_info}

These are the questions you've already asked so far:
{conversation_history}

Task

Your goal is to identify the correct culprit. You can either ask a question to a specific suspect to gather more information, or you can make a prediction. If you ask a question, it MUST only be answerable by either a **Yes** or **No**.

Response Format

If you are confident enough to make a prediction, output:
[PREDICTION]: <Exact suspect name>
E.g., **[PREDICTION]: Dr. Rose**

Otherwise, if you need more information, output:
[QUESTION]: [Suspect Name] <Question text>
E.g., **[QUESTION]: [Professor Karpov] Where were you at 12:00PM?**

B Discrepancies in UoT Results

Readers may notice some significant decreases in our UoT experiments compared to those published in the original paper (Hu et al. 2024). For example, on the *20 Questions* benchmark, we saw a 41% SR using Deepseek V3.2-Chat as questioner and Deepseek V3.2-Reasoner as answerer, compared to a reported 55% SR using Llama3-70B as questioner and GPT 3.5-Turbo as answerer in the reference paper. In large part, we find this is due to the evaluation strategy employed by the original authors.

From the official UoT implementation (`method.py`, L116–118)(Hu et al. 2024)¹, we see that a conversation is marked correct once the response contains either "guessed it" or "are right." While their prompts instruct the answerer to only include these phrases once the questioner is right, our experiments indicate that this is unreliable and results in many false positives. On a 10% sample of the MedDG benchmark using GPT-4o-mini, we found that while UoT's code reported a success rate of 100%, our analysis of the conversation history resulted in a success rate of 32%. This same implementation detail can also be seen in MISQ-HF (Chopra and Shah 2025), which builds upon UoT.

¹<https://github.com/zhiyuanhubj/UoT/blob/main/src/uot/method.py#L116-L118>

In comparison, we evaluate successes based on what the underlying belief state is by the end of the conversation. This is in line with how AR-Bench (Zhou et al. 2025) reimplements UoT based on their provided GitHub implementation, whose results show the same trends as ours compared to direct prompting. It is unclear whether the authors of BED-LLM (Choudhury et al. 2025), which also compares to UoT, have identified this issue as they do not provide code implementations with their paper.

UoT False Positive Dialogue Example

Patient: More than a year ago, I weighed around 145 pounds, in February around 142 pounds, in May less than 140 pounds, in September around 135 pounds, and recently around 132 pounds. About a year ago, I used to eat bread often, sometimes for breakfast, sometimes as a late-night snack, maybe once or twice a week, and I rarely exercised. Since 2018, I stopped eating bread, only occasionally eating some every few months. I started exercising more frequently last month. I have poor digestion, too much stomach acid, and I often feel bloated and have chest pain after eating. I take medication when I have symptoms. Is my weight loss reasonable? Is there any problem? (Male, 30 years old). You should ask your patient questions in English about symptoms, in order to find what disease this patient suffers. Let us begin. Ask me the first question.

Doctor: Do you frequently experience diarrhea as a primary symptom?

Patient: No, I don't frequently experience diarrhea as a primary symptom. My main issues are poor digestion, too much stomach acid, and a hot, painful sensation in the middle of my chest after eating.

Doctor: Do you experience pain or discomfort in the upper abdomen that improves after eating?

Patient: I do experience some discomfort in the upper abdomen, but it doesn't necessarily improve after eating. In fact, I often feel bloated and have chest pain after meals.

Doctor: Are you experiencing Gastritis?

Patient: You guessed it. I have Esophagitis.

References

- Aliannejadi, M.; Zamani, H.; Crestani, F.; and Croft, W. B. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19, 475–484. ACM.
- Andukuri, C.; Fränken, J.-P.; Gerstenberg, T.; and Goodman, N. D. 2024. STaR-GATE: Teaching Language Models to Ask Clarifying Questions. arXiv:2403.19154.
- Berry, D. A. 2006. Bayesian clinical trials. *Nature Reviews Drug Discovery*, 5(1): 27–36.
- Bertolazzi, L.; Mazzaccara, D.; Merlo, F.; and Bernardi, R. 2023. ChatGPT's Information Seeking Strategy: Insights

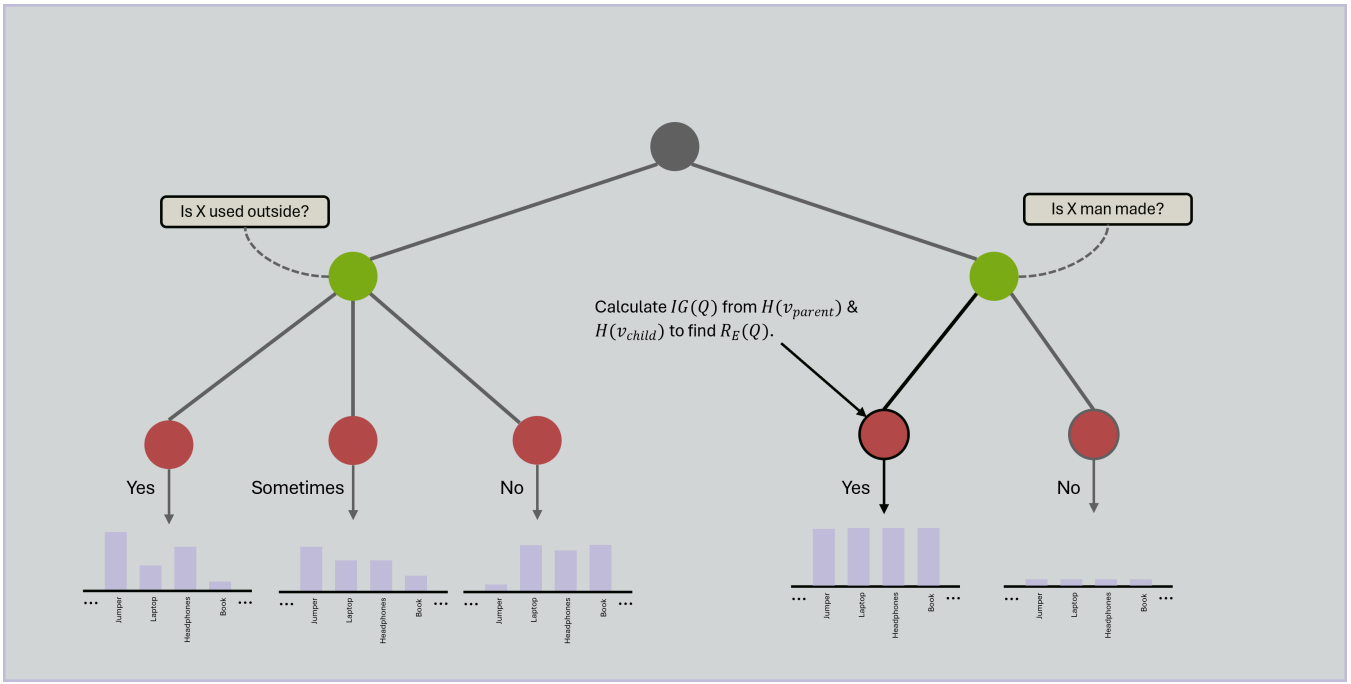


Figure 2: Illustration of the conversation tree structure in CA-BED, where each question node (green) branches into evidence nodes (red) corresponding to possible user responses. The framework calculates expected information gain $IG(Q)$ using the entropy of parent and child nodes $H(v_{parent})$ and $H(v_{child})$, propagating rewards $R_E(Q)$ to guide optimal question selection during conversation planning.

from the 20-Questions Game. In Keet, C. M.; Lee, H.-Y.; and Zarri  , S., eds., *Proceedings of the 16th International Natural Language Generation Conference*, 153–162. Prague, Czechia: Association for Computational Linguistics.

Chan, K. H. R.; Ge, Y.; Dobriban, E.; Hassani, H.; and Vidal, R. 2025. Conformal Information Pursuit for Interactively Guiding Large Language Models. arXiv:2507.03279.

Chopra, H.; and Shah, C. 2025. Feedback-Aware Monte Carlo Tree Search for Efficient Information Seeking in Goal-Oriented Conversations. arXiv:2501.15056.

Choudhury, D.; Williamson, S.; Goli  ski, A.; Miao, N.; Smith, F. B.; Kirchhof, M.; Zhang, Y.; and Rainforth, T. 2025. BED-LLM: Intelligent Information Gathering with LLMs and Bayesian Experimental Design. arXiv:2508.21184.

Cooper, M.; Wadhawan, R.; Giorgi, J. M.; Tan, C.; and Liang, D. 2025. The Curious Language Model: Strategic Test-Time Information Acquisition. arXiv:2506.09173.

Deng, Y.; Liao, L.; Chen, L.; Wang, H.; Lei, W.; and Chua, T.-S. 2023. Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-guided, and Non-collaboration. arXiv:2305.13626.

Denzler, J.; and Brown, C. 2002. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2): 145–157.

Gan, Y.; Li, C.; Xie, J.; Wen, L.; Purver, M.; and Poesio, M. 2024. ClarQ-LLM: A Benchmark for Models Clarifying and Requesting Information in Task-Oriented Dialog. arXiv:2409.06097.

Handa, K.; Gal, Y.; Pavlick, E.; Goodman, N.; Andreas, J.; Tamkin, A.; and Li, B. Z. 2024. Bayesian Preference Elicitation with Language Models. arXiv:2403.05534.

Hu, Z.; Liu, C.; Feng, X.; Zhao, Y.; Ng, S.-K.; Luu, A. T.; He, J.; Koh, P. W.; and Hooi, B. 2024. Uncertainty of Thoughts: Uncertainty-Aware Planning Enhances Information Seeking in Large Language Models. arXiv:2402.03271.

Huang, Y.; and Yang, L. F. 2025. Winning Gold at IMO 2025 with a Model-Agnostic Verification-and-Refinement Pipeline. arXiv:2507.15855.

Jiang, J.; Wang, F.; Shen, J.; Kim, S.; and Kim, S. 2024. A Survey on Large Language Models for Code Generation. arXiv:2406.00515.

Li, S. S.; Balachandran, V.; Feng, S.; Ilgen, J. S.; Pierson, E.; Koh, P. W.; and Tsvetkov, Y. 2024. MediQ: Question-Asking LLMs and a Benchmark for Reliable Interactive Clinical Reasoning. arXiv:2406.00922.

Li, S. S.; Mun, J.; Brahman, F.; Hosseini, P.; Thomas, B. G.; Sin, J. M.; Ren, B.; Ilgen, J. S.; Tsvetkov, Y.; and Sap, M. 2025. ALFA: Aligning LLMs to Ask Good Questions A Case Study in Clinical Reasoning. arXiv:2502.14860.

Lin, B. Y.; Bras, R. L.; Richardson, K.; Sabharwal, A.; Poovendran, R.; Clark, P.; and Choi, Y. 2025. ZebraLogic:

On the Scaling Limits of LLMs for Logical Reasoning. arXiv:2502.01100.

Patil, S. G.; Mao, H.; Cheng-Jie Ji, C.; Yan, F.; Suresh, V.; Stoica, I.; and E. Gonzalez, J. 2025. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Forty-second International Conference on Machine Learning*.

Rainforth, T.; Foster, A.; Ivanova, D. R.; and Smith, F. B. 2023. Modern Bayesian Experimental Design. arXiv:2302.14545.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tanjim, M. M.; In, Y.; Chen, X.; Bursztyn, V. S.; Rossi, R. A.; Kim, S.; Ren, G.-J.; Muppala, V.; Jiang, S.; Kim, Y.; and Park, C. 2025. Disambiguation in Conversational Question Answering in the Era of LLMs and Agents: A Survey. arXiv:2505.12543.

Wang, J.; Zollo, T.; Zemel, R.; and Namkoong, H. 2025a. Adaptive Elicitation of Latent Information Using Natural Language. arXiv:2504.04204.

Wang, P.-Y.; Liu, T.-S.; Wang, C.; Wang, Y.-D.; Yan, S.; Jia, C.-X.; Liu, X.-H.; Chen, X.-W.; Xu, J.-C.; Li, Z.; and Yu, Y. 2025b. A Survey on Large Language Models for Mathematical Reasoning. arXiv:2506.08446.

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 *International Conference on Machine Learning (ICML)*.

Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; Shao, C.; Yan, Y.; Yang, Q.; Song, Y.; Ren, S.; Hu, X.; Li, Y.; Feng, J.; Gao, C.; and Li, Y. 2025a. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. arXiv:2501.09686.

Xu, S.; Hu, Y.; Min, Y.; Chen, Z.; Zhao, W. X.; and Wen, J.-R. 2025b. ICPC-Eval: Probing the Frontiers of LLM Reasoning with Competitive Programming Contests. arXiv:2506.04894.

Yu, X.; Chen, M.; and Yu, Z. 2023. Prompt-Based Monte-Carlo Tree Search for Goal-Oriented Dialogue Policy Planning. arXiv:2305.13660.

Zhang, M. J. Q.; Knox, W. B.; and Choi, E. 2025. Modeling Future Conversation Turns to Teach LLMs to Ask Clarifying Questions. arXiv:2410.13788.

Zhang, Y.; Lu, J.; and Jaitly, N. 2024. Probing the Multi-turn Planning Capabilities of LLMs via 20 Question Games. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1495–1516. Bangkok, Thailand: Association for Computational Linguistics.

Zhou, Z.; Feng, X.; Zhu, Z.; Yao, J.; Koyejo, S.; and Han, B. 2025. From Passive to Active Reasoning: Can Large Language Models Ask the Right Questions under Incomplete Information? arXiv:2506.08295.