BEYOND FINDING FACTS: DISCOBENCH FOR EVALUATING CLARIFICATION IN DEEP SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Search agents powered by large language models (LLMs) increasingly tackle complex information-seeking tasks through multi-step retrieval and reasoning. However, existing benchmarks often assume complete and explicit queries, leaving agents' ability to clarify underspecified requests insufficiently evaluated. In multi-step search, retrieved facts may support several plausible intermediate targets without revealing which one the user intends. Final-answer correctness alone offers limited insight into whether agents recognize and resolve these information gaps. We introduce DISCOBENCH, a benchmark for evaluating whether search agents recognize clarification needs, ask effective questions, and use feedback to advance through interdependent retrieval steps. Starting from verified multi-hop questions, we introduce ambiguous checkpoints with annotated discriminative clues and design a user simulator that provides these clues in response to clarification questions. The annotated checkpoints provide verifiable intermediate targets for assessing whether agents ask effective clarification questions and successfully advance after receiving feedback. DISCOBENCH contains 211 samples and 463 ambiguity instances across 11 knowledge domains and four ambiguity types. We evaluate task utility, ambiguity detection, clarification effectiveness, and interaction cost. Experiments on 12 LLMs show that even the best-performing model achieves only 63.7% task accuracy, highlighting the challenge of coordinating retrieval with proactive clarification in multi-step search.

1 INTRODUCTION

In recent years, search agents based on Large Language Models (LLMs) have made significant progress in the field of information retrieval (Mialon et al., 2024; Wei et al., 2025; Wong et al., 2026). The search paradigm is shifting from traditional static corpus retrieval to autonomous Web Search Agents capable of handling complex goals (Xi et al., 2026; OpenAI, 2025; Google, 2025). These agents can simulate human navigation and browsing behaviors, achieving multi-step reasoning and information integration in dynamic and complex internet environments (Wu et al., 2025; Zhou et al., 2024).

However, current paradigms often presuppose that the user's initial query is complete and explicit. In real-world information seeking, users may provide vague or incomplete queries due to blurred memories or cognitive load limitations (Aliannejadi et al., 2019; Zamani et al., 2020). In deep search scenarios, such underspecification can affect multiple stages of a task. Unlike traditional single-hop retrieval (Kwiatkowski et al., 2019), deep search involves multi-step reasoning chains (Trivedi et al., 2022), where identifying an intermediate entity or constraint can be necessary for subsequent retrieval. As illustrated in Fig. 1, ambiguity at an intermediate step can leave the intended target unresolved and hinder completion of the remaining task. Therefore, interactive clarification provides an important means of obtaining missing information. Recent model evaluations examine whether agents recognize informational limits and avoid unsupported actions (OpenAI, 2026; Anthropic, 2026), motivating assessment of how agents address information gaps during multi-step search. However, final-answer correctness alone makes it difficult to determine whether an agent initiates clarification when needed and whether that clarification helps it complete subsequent retrieval. Evaluation therefore needs to examine individual retrieval steps: specifying what information remains unresolved and which supplementary clues can resolve the ambiguity, and assessing whether the agent can ask targeted questions and use the feedback to complete the task.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

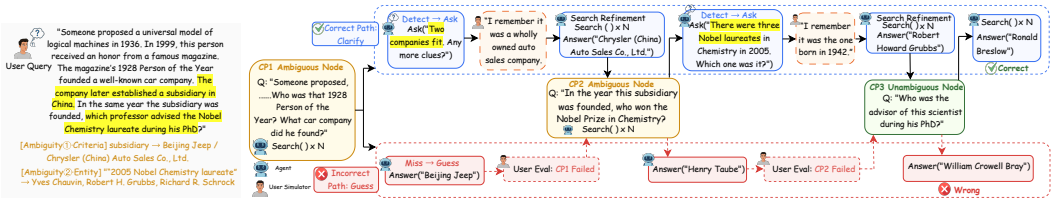

Fig. 1: A motivating example of unresolved intermediate targets in interactive deep search.

While the academic community has recognized the importance of query ambiguity and interaction, existing benchmarks still struggle to fully evaluate the disambiguation capabilities of search agents. Mainstream retrieval benchmarks (e.g., GAIA (Mialon et al., 2024), BrowseComp (Wei et al., 2025), AgentBench (Liu et al., 2024)) mostly assume explicit queries and focus on multi-hop reasoning, with less attention to proactive clarification. Ambiguity-focused datasets (e.g., AmbigQA (Min et al., 2020), DEEPAMBIGQA (Ji et al., 2025)) primarily consist of static scenarios and lack dynamic interaction simulation, whereas interaction-based benchmarks (e.g., IN3 (Qian et al., 2024), UserBench (Qian et al., 2025)) offer limited coverage of clarification in open-domain multi-step web retrieval. The recent INTERACTCOMP (Deng et al., 2026) has begun to address search interaction. Building on this direction, we focus on checkpoint-level diagnosis across multi-step retrieval tasks, with explicitly annotated ambiguity and discriminative clues.

We introduce **DISCOBENCH** (**D**eep **I**nteractive **S**earch with **C**larificati**O**n Benchmark), which evaluates whether search agents can proactively clarify and resolve ambiguity during multi-step web search. Starting from verified multi-hop questions, we introduce ambiguous checkpoints within interdependent retrieval steps and associate each with discriminative clues. Agents must decide when to continue searching, request clarification, or submit an answer, while a simulated user provides the clues in response to their clarification questions. By explicitly annotating ambiguity and the information needed to resolve it, DiscoBench enables a fine-grained evaluation of whether agents recognize clarification needs, ask effective questions, and use feedback to advance subsequent steps.

Our central question is whether agents can recognize information gaps, ask targeted questions, and use feedback to advance the task. We evaluate 12 representative LLMs, with the best-performing model achieving only 63.7% task accuracy. Current models leave many clarification needs unresolved at the ambiguous checkpoints they reach. Behavioral analysis further shows that, across all evaluated models, retrieval followed by clarification is associated with higher checkpoint success than retrieval followed by answering without clarification. These results highlight the importance of deciding when to ask effective clarification questions alongside acquiring evidence in multi-step search.

Our main contributions are as follows:

- We construct DISCOBENCH, comprising 211 multi-step retrieval questions and 463 annotated ambiguity instances across 11 knowledge domains and four ambiguity types, to evaluate proactive clarification across interdependent retrieval steps.
- We introduce a checkpoint-level interactive evaluation framework that combines ambiguity and discriminative-clue annotations with a user simulator providing clues on request, jointly assessing task utility, ambiguity detection, clarification effectiveness, and interaction cost.
- We evaluate 12 representative LLMs, revealing shortcomings in task completion and targeted-clarification coverage. Behavioral analysis further identifies a consistent association between clarification after retrieval and higher checkpoint success.

2 RELATED WORK

2.1 WEB SEARCH BENCHMARK

Efforts to benchmark search agents often bifurcate into two dimensions. One branch focuses on reasoning depth, challenging agents to navigate complex web hierarchies for multi-hop tasks, such as GAIA (Mialon et al., 2024) and the BrowseComp series (Wei et al., 2025; Zhou et al., 2025a; Chen et al., 2026). In parallel, other benchmarks explore information width, necessitating the

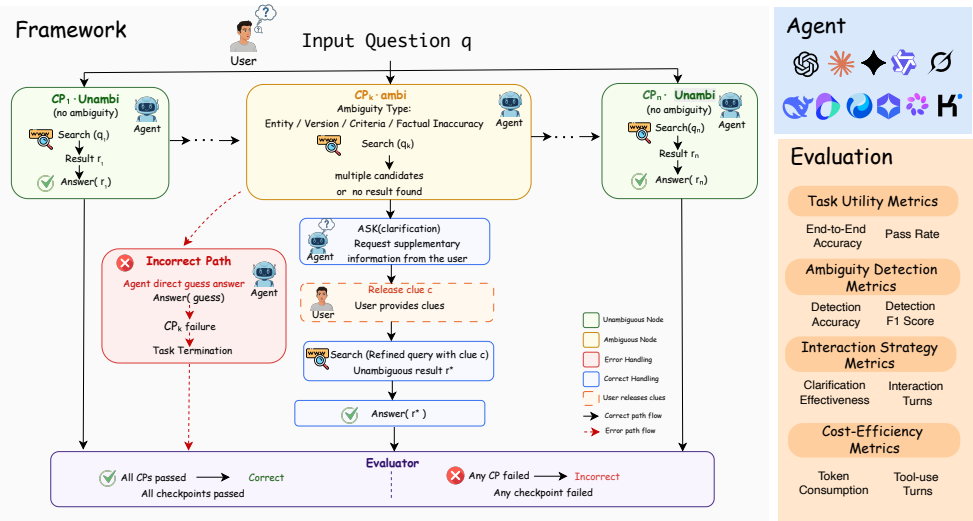

Fig. 2: Overview of the proposed interactive retrieval framework and evaluation protocol.

synthesis of vast horizontal data, as seen in PaSa (He et al., 2025), SPAR (Shi et al., 2025), and WideSearch (Wong et al., 2026). Complementary agent benchmarks like WebArena (Zhou et al., 2024), VisualWebArena (Koh et al., 2024), Mind2Web (Deng et al., 2023), and WebShop (Yao et al., 2022) further evaluate web navigation capabilities in realistic environments. Recent work, such as DeepWideSearch (Lan et al., 2025), has further begun to encompass both dimensions. Constrained by the assumption of complete queries, these benchmarks leave query ambiguity unexplored, which remains an essential aspect of autonomous search.

2.2 Ambiguity Benchmark

Research on query ambiguity has evolved into several taxonomies. Semantic and structural ambiguity benchmarks represented by AmbiEnt (Liu et al., 2023) investigate logical divergences arising from linguistic properties. In the realm of multi-answer factual ambiguity, datasets such as AmbigQA (Min et al., 2020) and ASQA (Stelmakh et al., 2022) focus on mapping single queries to multiple concurrent valid facts. Furthermore, conditional and contextual ambiguity benchmarks including TempAmbigQA (Piryani et al., 2024), CondAmbigQA (Li et al., 2025) and SituatedQA (Zhang & Choi, 2021) address scenarios where answers depend on latent temporal or geographical backgrounds that remain unstated in the initial query. While valuable, these benchmarks rely on a static evaluation paradigm that prioritizes answer identification over the dynamic, interactive process required for agent-user collaboration to resolve uncertainty.

2.3 Interactive Clarification Benchmark

Interactive benchmarks evaluate agent performance in multi-turn collaborative environments. For instance, ColBench (Zhou et al., 2025b) and UserBench (Qian et al., 2025) target code generation and travel planning, respectively, while IN3 (Qian et al., 2024) and GAIA2 (Froger et al., 2026) investigate implicit intent understanding and conflicting requests in local environments. In conversational QA, Abg-CoQA (Guo et al., 2021) requires agents to clarify coreference or semantic vagueness. IDRBench (Feng et al., 2026) supports autonomous web exploration and interactive deep research, evaluating long-form report quality, intent alignment, and interaction costs; we focus on retrieval checkpoints with verifiable answers. InteractComp (Deng et al., 2026) evaluates interactive disambiguation using target–distractor ambiguity and hidden context. DiscoBench adds annotated checkpoints across interdependent retrieval steps, providing verifiable intermediate targets for assessing effective clarification and subsequent task advancement.

3 TASK FORMULATION

We formulate multi-turn interactive retrieval as a sequential question-answering task in which an agent resolves a complex question q through a series of structured checkpoints(CP), with the ability to interact with a user when ambiguity is encountered. As illustrated in Fig. 2, the agent must determine whether the current retrieval state is ambiguous and decide whether to continue retrieval or request clarification from the user.

Question and Checkpoints. The protocol decomposes each question q in advance into an ordered sequence of n checkpoints $\{CP_1, CP_2, \ldots, CP_n\}$, each representing an intermediate retrieval sub-goal. A checkpoint CP_i is assigned one of two types:

- **Unambi**: an unambiguous checkpoint where the agent can answer directly via retrieval.
- **Ambi**: an ambiguous checkpoint containing one of four injected ambiguity types, which causes retrieval to return multiple candidates or no valid result.

Rationale for Checkpoint-Based Evaluation. Autonomous search permits different decompositions and retrieval paths. Final answers alone do not locate intermediate failures, while comparing open-ended trajectories requires additional annotation and step alignment. We therefore organize verified multi-hop questions into interdependent checkpoints with annotated sub-goals, disclosed information, ambiguities, and discriminative clues. Progressive disclosure ties clarification decisions to explicit information states and provides repeatable evaluation references for effective questioning and task advancement, without redefining intermediate assessment targets for each agent-generated trajectory.

Agent Actions and User Interaction. At each checkpoint CP_i, the agent may execute one of three actions:

$$a_i \in \{\textsc{Search},\ \textsc{Ask},\ \textsc{Answer}\} \tag{1}$$

For unambiguous checkpoints, the agent directly issues $\textsc{Search}(q_i)$ and proceeds with $\textsc{Answer}(r_i)$. For an ambiguous checkpoint CP_k, the agent should invoke $\textsc{Ask}(\cdot)$ to request supplementary information, upon which the user releases a pre-defined clue c. The agent then refines its search and resolves the ambiguity before issuing $\textsc{Answer}(r^*)$. In the Neutral setting, ambiguity labels, candidate sets, and ambiguity logic are hidden from the agent; questions are evaluated by their semantic target rather than prescribed wording. The harness supplies standard sub-questions, confirms correct answers, and stops at the first error, diagnosing clarification and progress under a supplied decomposition.

4 METHODOLOGY OF DATASET CONSTRUCTION

We construct DISCOBENCH to evaluate whether LLM agents can identify ambiguity, ask effective clarification questions, and use feedback to advance through interdependent retrieval steps. As illustrated in Fig. 3, the semi-automatic pipeline first builds verifiable multi-hop seed questions and then injects ambiguity with discriminative facts for interactive disambiguation.

4.1 SEED DATA PREPARATION

Topic & Seed Collection. We manually collect seed topics from 11 knowledge domains and use encyclopedic resources (e.g., Wikipedia and Baidu Baike) together with search results from Google, Bing, and Baidu. DISCOBENCH is currently constructed in Chinese to reflect ambiguity patterns and retrieval behaviors in Chinese web environments. We retain only questions with objectively verifiable answers that require external retrieval rather than common-sense reasoning alone.

Multi-hop QA Construction. Inspired by existing multi-hop QA datasets (Ho et al., 2020; Trivedi et al., 2022), we combine LLM-based expansion with human verification. LLMs generate single-hop factual QA pairs and expand them with retrieved evidence into candidate multi-hop chains. Human annotators then review the chains for factual correctness, retrieval feasibility, and logical coherence.

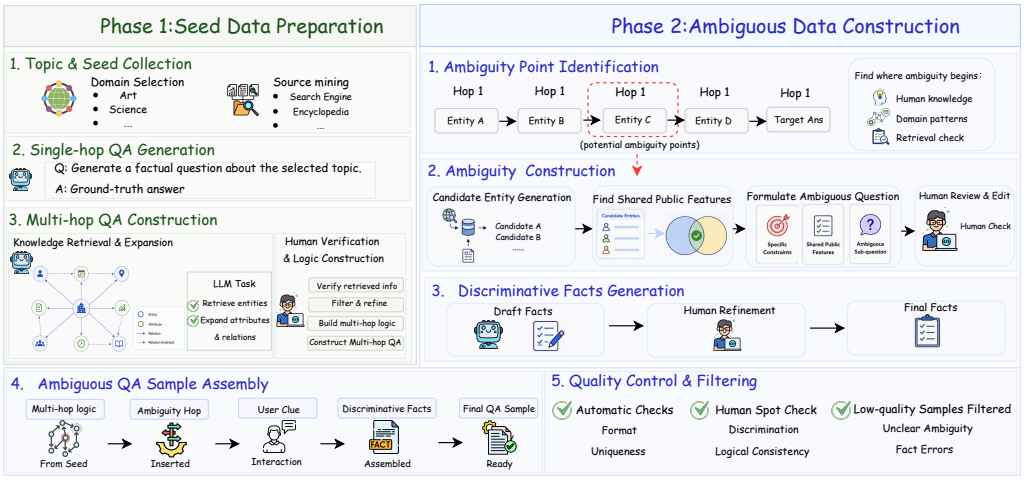

Fig. 3: Overview of the two-phase dataset construction pipeline, including seed multi-hop QA construction, ambiguity injection, discriminative fact generation, and quality control.

4.2 Ambiguous Data Construction

Ambiguity Point Identification. Given a deterministic multi-hop reasoning chain, we identify hops where ambiguity can be naturally introduced. Instead of injecting ambiguity randomly, we focus on nodes whose target entity has similar alternatives, such that relaxing the distinguishing constraint leads to multiple plausible candidates. A node is retained as a candidate ambiguity point if: (1) its target entity shares attributes with sibling entities; (2) the downstream reasoning chain remains executable under underspecification; and (3) the ambiguity can be resolved with a single user-provided clue. All candidate positions are further verified manually.

Ambiguity Construction. After identifying ambiguity points, we replace strong constraints with shared attributes among retrieved candidate entities, allowing multiple candidates to satisfy the rewritten description. Controlled injection ensures that each ambiguous state is auditable, retrieval-grounded, and paired with a minimal discriminative clue and a verifiable gold answer. Human annotators further verify naturalness, solvability, and logical consistency.

Discriminative Facts Generation. To support interactive disambiguation, we construct discriminative facts for each ambiguity point, which simulate supplementary clues provided by users and distinguish the target entity from distractors. Candidate facts are generated by retrieval-augmented LLMs from perspectives such as entity attributes, temporal information, relations, numerical facts, version differences, and organizational associations, and are then manually verified for factual correctness, distinguishability, and naturalness.

4.3 Data Statistics and Quality Control

Tab. 1 presents the overall statistics of DiscoBench, including domain distribution, task difficulty, and ambiguity types. Task difficulty is determined by the number of ambiguity checkpoints, where easy, medium, and hard correspond to 1, 2, and 3 ambiguity points, respectively. DiscoBench covers four ambiguity types: *Entity* (multiple entities satisfy the same description), *Version* (different temporal or version-specific states), *Criteria* (different evaluation standards or ranking criteria), and *Factual Inaccuracy* (descriptions inconsistent with objective facts).

An expert instructor, six undergraduate annotators from diverse disciplines, and two quality inspectors constructed the dataset. All samples were reviewed for factual correctness, retrieval feasibility, logical consistency, and ambiguity solvability.

Tab. 1: Statistics of DISCOBENCH. Entries give counts and percentages in parentheses, computed over 211 samples for difficulty and domain and 463 instances for ambiguity type.

Difficulty			**Ambiguity**			
Easy	Medium	Hard	Entity	Factual Inac.	Version	Criteria
44 (20.9%)	82 (38.9%)	85 (40.3%)	176 (38.0%)	125 (27.0%)	109 (23.5%)	53 (11.4%)

Domain					
Film/TV	32 (15.2%)	Finance	18 (8.5%)	Medicine	16 (7.6%)
Games	27 (12.8%)	Sports	18 (8.5%)	Technology	15 (7.1%)
Academic	22 (10.4%)	Geography	18 (8.5%)	Policy/Law	10 (4.7%)
Art	19 (9.0%)	Music	16 (7.6%)		

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and Tools. We evaluate the 12 models in Tab. 2 under a shared retrieval framework and checkpoint evaluator, using the highest available reasoning effort where configurable. Appendix E lists API configurations. All SEARCH calls use Tavily (Tavily Inc., 2026), with retrieval reproducibility measures described in Appendix G.

Evaluation Harness. We use Gemini-3.5-Flash as a single state-aware responder for simulated-user replies, checkpoint decisions, and final-answer decisions. At the final checkpoint, it compares the agent answer with the provided gold answer. All reported experimental results are averages over repeated runs. Appendix K reports agreement with human judgments for final-answer grading and checkpoint decisions.

Metrics. We report metrics from four aspects. For task utility, we use end-to-end accuracy and checkpoint pass rate. For ambiguity detection, we report detection accuracy, detection F1, and clarification recall, which measures correctly targeted clarification over reached ambiguous checkpoints. For interaction quality, we report the accuracy of the clarification question (CE-A), the clarification-to-advance rate (CE-B), and over-asking rate, which exposes unnecessary questions at non-ambiguous checkpoints. For cost efficiency, we report average ask turns, tool-use turns, and token consumption. Detailed definitions of all metrics and additional analysis of token consumption are provided in Appendix A and Appendix F.

5.2 MAIN RESULTS

Current models struggle with multi-step search that requires proactive clarification. Tab. 2 shows that the best-performing model, GPT-6-ASTRA, achieves 63.7% task accuracy, leaving more than one-third of tasks incomplete. Eight of the 12 models score below 50%. Clarification recall ranges from 39.0% to 78.5%, indicating that even the strongest model does not ask effective clarification questions at all ambiguous checkpoints it reaches.

The strong task performance of GPT-6-ASTRA and Claude-Opus-5 is qualitatively consistent with official evaluations of acknowledging tool limitations (OpenAI, 2026) and verifying assumptions before acting (Anthropic, 2026), respectively. DISCOBENCH examines related behavior when resolving information gaps through user clarification.

Successful clarification when asking can coexist with unmet clarification needs. DeepSeek-V4.1-Flash achieves 94.4% CE-B but only 41.9% clarification recall. The former means that, among checkpoints where it asks, 94.4% involve correctly targeted clarification followed by successful advancement. The latter means that only 41.9% of the ambiguous checkpoints it reaches receive correctly targeted questions. Ambiguous checkpoints where it does not ask reduce recall but are excluded from CE-B, so strong performance on the questions it chooses to ask can coexist with

Tab. 2: Main results under Neutral prompting. Values are percentages except average ask turns (Ask). Acc.: task accuracy. CP: checkpoint pass rate. Det.: ambiguity detection. Recall: clarification recall. CE-A/B: clarification quality. Darker blue indicates better performance. Lower over-asking is better.

Model	Acc.	CP	Det. Acc.	Det. F1	Recall	Over-ask	CE-A	CE-B	Ask
GPT-6-ASTRA	**63.7**	**74.9**	**84.2**	**83.8**	**78.5**	9.6	89.1	85.6	2.05
Claude-Fable-5.1	55.5	73.1	78.8	76.7	66.0	6.7	91.0	89.4	1.25
Claude-Opus-5	55.0	69.1	74.7	70.9	56.7	3.8	93.8	90.0	1.14
Muse-Spark-1.3	54.6	70.9	80.5	78.7	68.0	5.4	91.4	86.8	1.34
Hunyuan-4-Preview	46.4	62.1	74.5	70.7	57.2	5.5	92.0	86.9	0.88
Gemini-3.8-Flash	46.1	64.3	75.1	71.9	58.1	4.3	93.0	89.3	1.19
Grok-4.6	41.9	59.8	78.2	76.5	65.6	7.0	89.4	83.0	1.51
DeepSeek-V4.1-Flash	38.7	58.5	67.9	58.7	41.9	**0.8**	**96.8**	**94.4**	0.69
GLM-5.3-Flash	38.2	56.1	70.6	65.0	49.5	3.5	94.0	88.0	0.92
Kimi-K3	36.8	55.8	64.6	55.6	39.3	2.6	94.4	92.1	0.42
Qwen3.8-Max	34.6	56.9	66.2	57.6	41.5	3.3	94.8	92.2	0.74
Doubao-Seed-2.1-Pro	28.4	49.7	64.3	55.3	39.0	2.7	95.0	93.4	0.66

many clarification needs left unaddressed. GPT-6-ASTRA has lower CE-B (85.6%) but higher recall (78.5%), covering a larger share of the ambiguity it encounters.

Higher clarification recall does not ensure higher task accuracy. Grok-4.6 has higher recall than Claude-Opus-5 (65.6% versus 56.7%), but lower CE-B (83.0% versus 90.0%) and task accuracy (41.9% versus 55.0%). Coverage alone is insufficient: models must also ask effective questions and use the answers to advance, motivating joint interpretation of recall and CE-B.

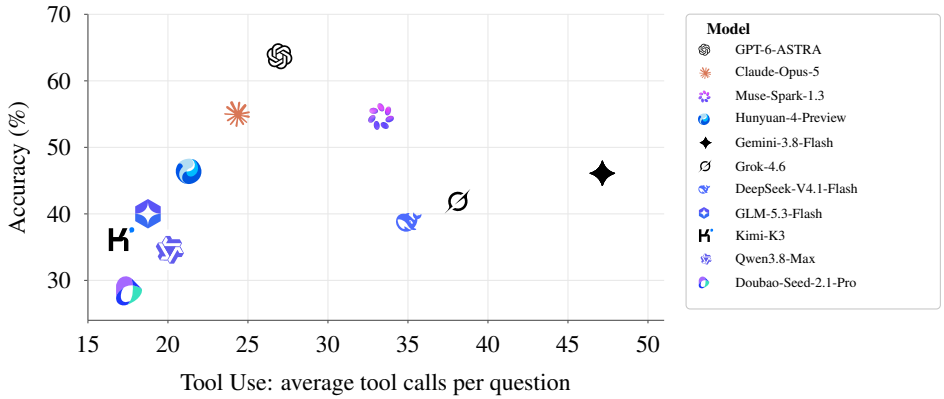

Fig. 4: Performance–efficiency trade-off under neutral prompting.

Retrieval volume does not measure resolution of user intent. Fig. 4 shows that Gemini-3.8-Flash uses 47.2 tool calls per question, compared with GPT-6-ASTRA's 27.0, yet achieves lower accuracy (46.1% versus 63.7%). Search may support several plausible candidates without supplying the user-specific constraint that distinguishes them, requiring agents to decide when to seek clarification rather than more evidence. Sec. 5.5 examines this choice after retrieval. Appendix L evaluates responder-prompt sensitivity, and Appendix M reports ranking stability, repeated-run variability, and evaluation cost.

5.3 Effect of Guided Prompting on Selected Models

Under Neutral prompting, the agent receives no explicit reminder of potential ambiguity and decides independently whether to ask for clarification. Guided prompting explicitly reminds the agent to consider potential ambiguity and ask clarification questions when needed.

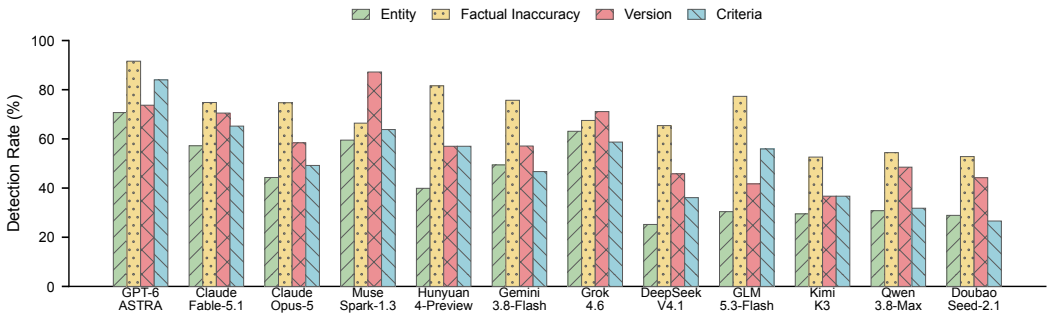

Fig. 5: Clarification recall by ambiguity type under Neutral prompting, computed over reached ambiguous checkpoints.

Guided prompting increases both clarification recall and over-asking in five of the six models. Accuracy improves by 5.2–15.3 percentage points for Gemini, DeepSeek, Hunyuan, and Doubao, while Claude-Opus-5 remains nearly unchanged. GPT-6-ASTRA instead loses 3.7 accuracy points despite higher recall and substantially more asks. Thus, explicit guidance can increase targeted-clarification coverage, but more frequent interaction does not consistently translate into better task completion. Appendix B reports the full paired results and interaction metrics.

5.4 Performance by Ambiguity Types

Fig. 5 shows that recall on reached checkpoints varies with the kind of missing information. Version recall exceeds Criteria recall for 8 of the 12 models. The largest gaps are 23.4 percentage points for Muse-Spark-1.3 and 17.6 points for Doubao-Seed-2.1-Pro. A plausible explanation is that competing versions expose concrete alternatives, whereas an unspecified criterion can leave a retrieved answer apparently valid without revealing which standard the user intends. The ordering is not universal: GPT-6-ASTRA achieves 84.0% recall on Criteria versus 73.7% on Version, and GLM-5.3-Flash achieves 55.9% on Criteria versus 41.7% on Version. GPT's recall ranges from 70.7% to 91.6% across the four types. This pattern motivates attention to implicit user constraints alongside explicit conflicts among retrieved candidates.

5.5 Behavioral Profile Analysis

Under the search-first Neutral protocol, we classify behavior at reached ambiguous checkpoints into DirectGuess, SearchThenGuess, DirectAsk, and SearchThenAsk. SearchThenGuess denotes retrieval followed by answering without clarification. Appendix D gives the classification rules and per-model distributions.

Models differ in how often they ask after retrieval. SearchThenAsk accounts for 40.6–75.4% of ambiguous checkpoints across models, whereas its pass rate ranges from 89.4% to 98.7%. Doubao-Seed-2.1-Pro asks after retrieval at 41.9% of reached ambiguous checkpoints, and 98.7% of those checkpoints pass. It therefore succeeds on this branch but selects it at fewer than half of reached ambiguous checkpoints.

Asking after retrieval is associated with higher checkpoint success. Tab. 3 shows higher pass rates for SearchThenAsk than SearchThenGuess in all 12 models, with gaps of 28.5–49.6 percentage points and unweighted means of 94.2% versus 57.0%.

5.6 Diagnostic Setting Comparisons

Interaction improves performance even with search available. In Tab. 4, No-Ask retains search but disables questions, scoring below Full for all four models by 10.9–31.5 percentage points. GPT-6-ASTRA improves from 32.2% to 63.7%, and Doubao-Seed-2.1-Pro from 17.5% to 28.4%. Search

Tab. 3: Checkpoint pass rates (%). Read down each block, ordered by STA share in Tab. 6.

Model	DG	STG	STA	Model	DG	STG	STA
GPT-6-ASTRA	50.0	61.4	**97.0**	Claude-Opus-5	–	65.1	**93.6**
Muse-Spark-1.3	–	57.0	**93.6**	GLM-5.3-Flash	100.0	57.9	**89.7**
Claude-Fable-5.1	–	63.3	**96.0**	DeepSeek-V4.1-Flash	33.3	58.1	**94.4**
Grok-4.6	100.0	53.4	**89.4**	Qwen3.8-Max	100.0	58.4	**94.7**
Gemini-3.8-Flash	50.0	56.0	**94.5**	Doubao-Seed-2.1-Pro	–	49.1	**98.7**
Hunyuan-4-Preview	70.0	55.6	**92.2**	Kimi-K3	64.3	48.4	**96.2**

DG: DirectGuess. STG: SearchThenGuess. STA: SearchThenAsk. –: no trajectory. Mean pass rates: STG 57.0%, STA **94.2**%.

Tab. 4: Accuracy (%) by setting. Δ: percentage-point change from the Full (Neutral) baseline.

Model	Full	No-Ask		Ask-only		Unambig.	
		Acc.	Δ	Acc.	Δ	Acc.	Δ
GPT-6-ASTRA	63.7	32.2	-31.5	3.2	-60.5	96.3	+32.6
Hunyuan-4-Preview	46.4	27.5	-18.9	9.4	-37.0	90.5	+44.1
DeepSeek-V4.1-Flash	38.7	24.2	-14.5	14.6	-24.1	87.4	+48.7
Doubao-Seed-2.1-Pro	28.4	17.5	-10.9	7.9	-20.5	86.8	+58.4

alone retains some ability to solve these tasks, but the consistent gap shows that access to retrieved evidence does not make user interaction redundant.

Interaction does not fully close the gap to explicit constraints. Doubao has 93.4% CE-B but only 39.0% targeted-clarification coverage. Full exceeds No-Ask by 10.9 points yet trails Unambiguous by 58.4 points. GPT's corresponding differences are 31.5 and 32.6 points. The remaining gaps suggest that obtaining and using missing constraints during retrieval remains challenging even when the model performs well after asking. Ask-only disables search and directs fact verification to user interaction, yielding 3.2–14.6% accuracy. Appendix N documents the configurations.

5.7 Robustness to Ambiguity Count and Reasoning Effort

Model rankings vary with ambiguity count. Appendix C presents ambiguity-count results in Fig. 6. Claude-Fable-5.1 exceeds GPT-6-ASTRA on Easy questions (73.0% versus 61.0%), but the ranking reverses on Hard questions (40.7% versus 55.2%). GPT's Easy-to-Hard drop is 5.8 points, compared with Fable's 32.3, suggesting stronger performance across successive ambiguity points. All models score lower on Hard than Easy, although GPT and Muse peak on Medium. These question groups reveal performance differences obscured by aggregate accuracy.

The value of higher reasoning effort is model-dependent. Fig. 7 in Appendix C compares GPT-6-ASTRA at max versus low effort and Gemini-3.8-Flash at high versus low. Lower effort roughly halves search calls for both models, but accuracy falls by only 4.2 points for GPT (63.7% to 59.5%) and by 17.2 points for Gemini (46.1% to 28.9%). Clarification recall also declines more for Gemini (24.3 versus 9.4 points). For GPT, higher reasoning effort improves both recall and accuracy, whereas Guided prompting raises recall while reducing accuracy.

6 Conclusion

DiscoBench diagnoses clarification across annotated retrieval checkpoints. Across 12 models, targeted-clarification coverage, conditional question quality, and task completion yield distinct performance profiles. Guided prompting often increases both coverage and over-asking, without consistent accuracy gains. Asking after retrieval is associated with higher checkpoint success. Together, these results show how checkpoint diagnostics complement final-answer accuracy in evaluating interactive search.

AI USE STATEMENT

In this work, we used generative AI tools to edit and polish the manuscript and to assist dataset construction. We have not used generative AI tools for experiment design or result interpretation, and mathematical claims and proofs are not applicable to this work. We have reviewed all AI-assisted work. Human annotators verified the generated items, and the authors checked the revised text against the experimental records. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

DISCOBENCH is constructed from publicly available web resources, including encyclopedic websites and search engine result pages, and does not involve private or personally identifiable information. The benchmark is designed solely for research purposes to evaluate ambiguity handling and clarification abilities in search agents.

REPRODUCIBILITY STATEMENT

The appendices provide the checkpoint evaluation rules (Appendix A), model identifiers and reasoning settings (Appendix E), prompt templates (Appendix N), harness validation and responder-prompt checks (Appendices K and L), repeated-run variability and evaluation cost (Appendix M), token consumption (Appendix F), and the treatment of the black-box search backend (Appendix G). We retain timestamped search logs, queries, model outputs, and agent trajectories. Subject to provider terms and data-release permissions, the planned release will include the benchmark items, construction records, cached retrieval responses, and these traces, so recorded runs can be audited and replayed against a fixed retrieval surface.

REFERENCES

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 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*, pp. 475–484, 2019.

Anthropic. Claude Opus 5 System Card. https://www.anthropic.com/claude-opus-5-system-card, 2026. Accessed: 2026-09-21.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, et al. BrowseComp-Plus: A fair and disentangled evaluation benchmark for deep search agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22349–22370, 2026. doi: 10.18653/v1/2026.acl-long.1023.

Mingyi Deng, Lijun Huang, Yani Fan, Fanqi Kong, Jiayi Zhang, Fashen Ren, Jinyi Bai, Fuzhen Yang, Dayi Miao, Zhaoyang Yu, Yifan Wu, et al. InteractComp: Evaluating search agents with ambiguous queries. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Yingchaojie Feng, Qiang Huang, Xiaoya Xie, Zhaorui Yang, Jun Yu, Wei Chen, and Anthony KH Tung. IDRBench: Benchmarking the interactive capabilities of deep research agents. *arXiv preprint arXiv:2601.06676*, 2026.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, et al. Gaia2: Benchmarking LLM agents on dynamic and asynchronous environments. In *The Fourteenth International Conference on Learning Representations*, 2026.

Google. Gemini deep research. https://gemini.google/overview/deep-research/, 2025. Accessed: 2026-04-13.

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

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. PaSa: An LLM agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11663–11679, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Jiabao Ji, Min Li, Priyanshu Kumar, Shiyu Chang, and Saloni Potdar. DeepAmbigQA: Ambiguous multi-hop questions for benchmarking LLM answer completeness. *arXiv preprint arXiv:2511.01323*, 2025.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Tian Lan, Bin Zhu, Qianghuai Jia, Junyang Ren, Haijun Li, Longyue Wang, Zhao Xu, Weihua Luo, and Kaifu Zhang. DeepWideSearch: Benchmarking depth and width in agentic information seeking. *arXiv preprint arXiv:2510.20168*, 2025.

Zongxi Li, Yang Li, Haoran Xie, and S Joe Qin. CondAmbigQA: A benchmark and dataset for conditional ambiguous question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2269–2288, 2025.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 790–807, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797, 2020.

OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025. Accessed: 2026-04-13.

OpenAI. GPT-6 Astra System Card. https://deploymentsafety.openai.com/gpt-6-astra/, 2026. Accessed: 2026-09-21.

Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, and Adam Jatowt. Detecting temporal ambiguity in questions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9620–9634, 2024.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. Tell me more! Towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1088–1113, 2024.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, et al. UserBench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*, 2025.

Xiaofeng Shi, Yuduo Li, Qian Kou, Longbin Yu, Jinxin Xie, and Hua Zhou. SPAR: Scholar paper retrieval with LLM-based agents for enhanced academic search. *arXiv preprint arXiv:2507.15245*, 2025.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, 2022.

Tavily Inc. Tavily docs. https://docs.tavily.com/welcome, 2026. Accessed: 2026-09-21.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. WideSearch: Benchmarking agentic broad info-seeking. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. WebWalker: Benchmarking LLMs in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305, 2025.

Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. A survey of large language model-based search agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8244–8279, 2026. doi: 10.18653/v1/2026.acl-long.374.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. AssistantBench: Can web agents solve realistic and time-consuming tasks? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8938–8968. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.505.

Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. Analyzing and learning from user interactions for search clarification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1181–1190, 2020.

Michael Zhang and Eunsol Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7371–7387, 2021.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, et al. BrowseComp-ZH: Benchmarking web browsing ability of large language models in Chinese. *arXiv preprint arXiv:2504.19314*, 2025a.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. SWEET-RL: Training multi-turn LLM agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025b.

A Evaluation Metrics

This section provides detailed definitions of the evaluation metrics used in our experiments. All metrics are computed at the question level or checkpoint level and then averaged over all valid questions for each model.

Protocol and information visibility. Evaluation uses an annotated task decomposition: the harness supplies standard sub-questions sequentially, confirms correct answers, and terminates the question at the first incorrect answer. The agent chooses search, clarification, or an answer given the disclosed sub-question and interaction history; undisclosed later constraints are not part of that input. An Ambi label therefore denotes missing discriminative information in the current disclosure state, not necessarily ambiguity when the full rewritten question is shown at once. The protocol diagnoses clarification and progress under a supplied decomposition, rather than autonomous decomposition, continued error propagation, or recovery on incorrect paths. Questions that terminate before the final checkpoint count as task failures.

Population of checkpoint metrics. Let $\mathcal{R}_m$ denote the checkpoints reached by model m. Detection, clarification quality, and behavioral profiles describe states at which the agent actually has a decision opportunity; Fig. 5 additionally groups these states by type. Unreached checkpoints have no observed clarification decision and are therefore excluded from detection denominators. Earlier failures are reflected in task accuracy over all 211 questions and CP, whose denominator includes every annotated checkpoint per question. This separates overall task progress from clarification behavior in experienced states. Because $\mathcal{R}_m$ varies with trajectories, conditional metrics are interpreted alongside task metrics; type contrasts also reflect the composition and positions of reached checkpoints.

A.1 End-to-End Accuracy

End-to-end accuracy evaluates whether the agent's final answer to the full question matches the ground-truth answer. Equivalence between the two answers is determined by an LLM-based answer-equivalence judge, which abstracts away surface-form variations such as transliterations, date formats, and list ordering.

For each question q, let $\hat{a}_q$ denote the agent's final answer and $a_q^\star$ the ground-truth answer, and let $\mathrm{equiv}(\hat{a}_q, a_q^\star) \in \{0, 1\}$ denote the judge's binary verdict, where 1 indicates that the two answers are judged equivalent and 0 otherwise. The per-question correctness indicator is defined as:

$$\mathrm{Acc}(q) = \mathrm{equiv}\big(\hat{a}_q,\, a_q^\star\big)\,. \tag{2}$$

The model-level end-to-end accuracy is computed by averaging over all valid questions:

$$\mathrm{Acc} = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{Acc}(q), \tag{3}$$

where Q denotes the set of valid evaluated questions. This question-level normalization ensures that each question contributes equally to the final score.

A.2 Checkpoint Pass Rate

Each question in DiscoBench is decomposed into a sequence of checkpoints. A checkpoint is counted as successfully advanced if the agent either answers the checkpoint correctly and proceeds to the next checkpoint, or correctly completes the final checkpoint. In our evaluator logs, this corresponds to one of two outcomes: (1) `correct_answer`, where the agent correctly resolves a regular checkpoint; (2) `missed_ambiguity_correct`, where the agent misses the ambiguity at an ambiguous checkpoint but still happens to answer it correctly.

For each question q, let N_q denote the ground-truth number of checkpoints, and let A_q denote the number of checkpoints that are successfully advanced. The checkpoint pass score for question q is defined as:

$$\text{CP}(q) = \frac{A_q}{N_q}. \tag{4}$$

The model-level checkpoint pass rate is computed by averaging over all valid questions:

$$\text{CP} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \text{CP}(q), \tag{5}$$

where $\mathcal{Q}$ denotes the set of valid evaluated questions. This question-level normalization ensures that each question contributes equally to the final score, regardless of how many checkpoints it contains.

A.3 Ambiguity Detection Metrics

We evaluate ambiguity detection at the checkpoint level. For each reached checkpoint, we compare the ground-truth checkpoint type with the agent's interaction behavior. A checkpoint is labeled as *Ambi* if it contains an injected ambiguity, and *Non-Ambi* otherwise. We define the four detection outcomes as follows:

- **True Positive (TP)**: the checkpoint is *Ambi*, and the agent correctly asks a clarification question targeting the ambiguity.
- **False Negative (FN)**: the checkpoint is *Ambi*, but the agent fails to ask or does not correctly target the ambiguity.
- **False Positive (FP)**: the checkpoint is *Non-Ambi*, but the agent unnecessarily asks for clarification.
- **True Negative (TN)**: the checkpoint is *Non-Ambi*, and the agent proceeds without asking for clarification.

Accordingly, $TP + FN$ corresponds to all reached ambiguous checkpoints, while $FP + TN$ corresponds to all reached non-ambiguous checkpoints. The total number of evaluated detection decisions is $TP + TN + FP + FN$.

Detection Accuracy. Detection accuracy measures the overall correctness of ambiguity detection decisions:

$$\text{Detection Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{6}$$

Detection F1 Score. We further report detection F1 to better account for the imbalance between ambiguous and non-ambiguous checkpoints. Precision and recall are defined as:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}. \tag{7}$$

The detection F1 score is computed as:

$$\text{Detection F1} = \frac{2 \cdot P \cdot R}{P + R}. \tag{8}$$

Here, recall measures the proportion of reached ambiguous checkpoints with correctly targeted clarification, while precision measures how often the agent's clarification decisions are correct. F1 provides a more robust measure when ambiguous and non-ambiguous checkpoints are unevenly distributed.

A.4 Clarification Effectiveness

Clarification effectiveness evaluates the quality and usefulness of the agent's clarification behavior. Both metrics share the same denominator: the number of checkpoints where the agent actively invokes ASK, regardless of whether the question is correct.

Let $\mathcal{C}$ denote the set of evaluated checkpoints, and let $\mathcal{C}_{\texttt{asked}}$ denote the set of checkpoints where the agent invokes ASK:

$$\mathcal{C}_{\texttt{asked}} = \{c \in \mathcal{C} : \texttt{asked}(c)\}. \tag{9}$$

We further define:

$$\mathcal{C}_{\texttt{right}} = \{c \in \mathcal{C}_{\texttt{asked}} : \texttt{asked_right}(c)\}, \tag{10}$$

$$\mathcal{C}_{\texttt{adv}} = \{c \in \mathcal{C}_{\texttt{asked}} : \texttt{cp_advanced}(c)\}. \tag{11}$$

CE-A: Clarification Question Accuracy. CE-A measures whether the agent asks the right clarification question when it decides to interact:

$$\text{CE-A} = \frac{|\mathcal{C}_{\texttt{right}}|}{|\mathcal{C}_{\texttt{asked}}|}. \tag{12}$$

CE-B: Clarification-to-Advance Rate. CE-B measures whether a correct clarification eventually helps the agent advance the current checkpoint:

$$\text{CE-B} = \frac{|\mathcal{C}_{\texttt{right}} \cap \mathcal{C}_{\texttt{adv}}|}{|\mathcal{C}_{\texttt{asked}}|}. \tag{13}$$

CE-A reflects whether the agent asks in the correct direction, while CE-B further evaluates whether the agent can use the returned clue to successfully advance the checkpoint.

A.5 Clarification Recall and Over-asking Rate

Clarification recall measures the coverage of correctly targeted clarification over reached ambiguous checkpoints:

$$\text{Clarification Recall} = \frac{TP}{TP + FN}. \tag{14}$$

It is the recall component of ambiguity detection. FN includes both not asking and asking without correctly targeting the ambiguity, so recall jointly reflects interaction behavior and question targeting rather than asking frequency alone. CE-A and CE-B instead condition on asked checkpoints, measuring correct targeting and correct targeting followed by advancement, respectively. Over-asking rate is the fraction of reached non-ambiguous checkpoints where the agent asks:

$$\text{Over-asking Rate} = \frac{FP}{FP + TN}. \tag{15}$$

Together, these metrics characterize targeted coverage, conditional question quality, subsequent advancement, and unnecessary interaction.

B Supplementary Results for Guided Prompting

Tab. 5 reports paired Neutral and Guided results for six representative models on the same 211 questions, including accuracy, clarification recall, over-asking, and average asks. Neutral baselines match the main comparison.

Tab. 5: Neutral versus Guided prompting for six selected models. Neu./Gui.: accuracy (%); Δ: percentage-point change. Recall and over-asking (%) and average asks are Neutral→Guided. Both prompts are evaluated on the same 211 questions. Neutral baselines match Tab. 2.

Model	Neu.	Gui.	Δ	Clar. Rec.	Over-ask	Asks
GPT-6-ASTRA	63.7	60.0	-3.7	78.5→88.0	9.6→25.6	2.05→5.25
Claude-Opus-5	55.0	55.1	+0.1	56.7→54.4	3.8→3.7	1.14→0.87
Hunyuan-4-Preview	46.4	60.0	+13.6	57.2→71.3	5.5→16.4	0.88→1.37
Gemini-3.8-Flash	46.1	51.3	+5.2	58.1→73.8	4.3→13.3	1.19→1.71
DeepSeek-V4.1-Flash	38.7	51.6	+12.9	41.9→74.6	0.8→14.2	0.69→1.54
Doubao-Seed-2.1-Pro	28.4	43.7	+15.3	39.0→50.9	2.7→5.5	0.66→0.83

C Additional Figures on Ambiguity Count and Reasoning Effort

Fig. 6 presents performance across ambiguity counts, and Fig. 7 presents the reasoning-effort comparison. Section 5.7 discusses both results.

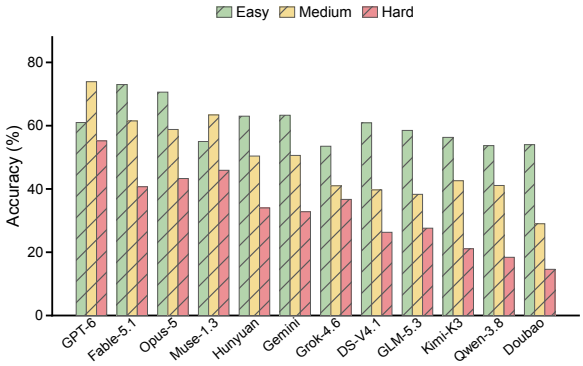

Fig. 6: Performance across ambiguity-complexity levels under neutral prompting.

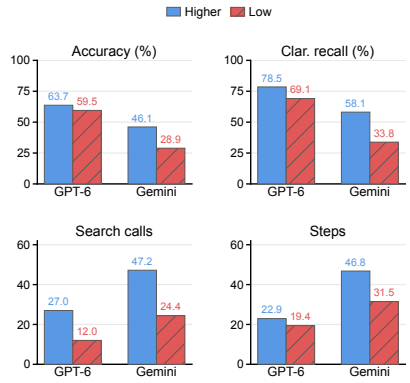

Fig. 7: Reasoning-effort comparison under Neutral prompting.

D Profile Classification Details

Behavioral profiles. We classify the first action segment of each ambiguous checkpoint, before any clarification clue is released, into four profiles:

- **DirectGuess**: no SEARCH, and the segment ends with ANSWER.
- **SearchThenGuess**: at least one SEARCH, and the segment ends with ANSWER.
- **DirectAsk**: no SEARCH, and the segment ends with ASK.
- **SearchThenAsk**: at least one SEARCH, and the segment ends with ASK.

SearchThenGuess corresponds to the `SearchHeavyGuess` label in the evaluation logs. We use the former name because the definition requires only one search, not a high search volume. Profile pass rates measure eventual checkpoint advancement after classification of the first segment; they are not success probabilities for an isolated first action.

DirectAsk rarity. DirectGuess and DirectAsk are rare. Their shares stay within 0–5.6% and 0–5.5%, respectively, and several models have neither profile. The Neutral prompt explicitly requests search first, and quality control filters some ambiguities recognizable without retrieval. These frequencies describe behavior under this protocol.

Profile distribution.

Tab. 6: Behavioral profile distribution (%) on ambiguous checkpoints. Models are sorted by STA ratio.

Model	DG	STG	DA	STA
GPT-6-ASTRA	1.2	17.9	5.5	**75.4**
Muse-Spark-1.3	0.0	30.3	0.4	**69.4**
Claude-Fable-5.1	0.0	31.8	0.0	**68.2**
Grok-4.6	0.3	32.6	0.3	**66.8**
Gemini-3.8-Flash	1.2	37.6	0.6	**60.6**
Hunyuan-4-Preview	3.8	36.3	0.0	**59.8**
Claude-Opus-5	0.0	41.4	0.0	**58.6**
GLM-5.3-Flash	1.3	46.3	0.0	**52.3**
DeepSeek-V4.1-Flash	1.1	53.6	0.7	**44.6**
Qwen3.8-Max	0.4	55.9	0.4	**43.4**
Doubao-Seed-2.1-Pro	0.0	58.1	0.0	**41.9**
Kimi-K3	5.6	53.0	0.8	**40.6**

DG denotes DirectGuess; STG denotes SearchThenGuess; DA denotes DirectAsk; STA denotes SearchThenAsk.

Conditional association. SearchThenAsk and SearchThenGuess pass rates summarize outcomes following model-selected actions at reached ambiguous checkpoints. They describe asking and answering branches within the retrieval workflow, with each profile evaluated on the checkpoints and histories occurring in its recorded trajectories.

E Evaluated Models and API Configurations

Tab. 7: Evaluated models and reasoning configurations. **Reasoning** shorthand: *xhigh / high / medium* – provider's discrete reasoning-effort or thinking-level setting; *thinking* – thinking/reasoning mode enabled (no effort granularity exposed); *adapt.* – adaptive thinking budget.

Model (paper)	Provider	API Identifier	Reasoning
GPT-6-ASTRA	OpenAI	`gpt-6-astra`	max
Claude-Fable-5.1	Anthropic	`claude-fable-5-1`	max
Claude-Opus-5	Anthropic	`claude-opus-5`	max
Muse-Spark-1.3	Meta	`muse-spark-1.3`	max
Hunyuan-4-Preview	Tencent	`hunyuan-4-preview`	high
Gemini-3.8-Flash	Google	`gemini-3.8-flash`	high
Grok-4.6	xAI	`grok-4.6`	xhigh
DeepSeek-V4.1-Flash	DeepSeek	`deepseek-v4.1-flash`	max
GLM-5.3-Flash	Zhipu AI	`glm-5.3-flash`	max
Kimi-K3	Moonshot AI	`kimi-k3`	max
Qwen3.8-Max	Alibaba	`qwen3.8-max-0902`	thinking
Doubao-Seed-2.1-Pro	ByteDance	`doubao-seed-2-1-pro-260915`	high
User simulator and checkpoint judge			
Gemini-3.5-Flash	Google	`gemini-3.5-flash`	medium

Tab. 7 lists the API configuration of every model used in our main experiments, restricted to the information needed to reproduce a call: the model name used in the paper, the provider, the API identifier used at invocation time, and the reasoning- or thinking-mode setting. To control for the confound of reasoning budget, models exposing a configurable effort level are run at the highest available setting in the main experiments. All SEARCH calls are executed through Tavily regardless of the agent backbone. A single auxiliary model, Gemini-3.5-Flash running with thinking level set to `medium`, serves both as the simulated user that releases discriminative clues during clarification turns and as the checkpoint-level judge that scores each step; its configuration is listed in the bottom block of Tab. 7 and uses a concurrency of 100.

F Token Consumption

To provide a supplementary reference for inference cost, we report the token consumption of the evaluated models under neutral prompting. Tab. 8 summarizes input and output tokens. Tab. 9 reports the models in the Section 5.6 experiment under Full, Ask-only, and Unambiguous settings.

Tab. 8: Token consumption of evaluated models under neutral prompting.

Model	Input	Output
GPT-6-ASTRA	2,463,091	9,877
Claude-Fable-5.1	2,809,128	16,843
Claude-Opus-5	1,942,589	17,865
Muse-Spark-1.3	2,067,081	29,678
Hunyuan-4-Preview	2,138,240	13,298
Gemini-3.8-Flash	9,700,059	22,804
Grok-4.6	1,990,812	14,673
DeepSeek-V4.1-Flash	2,712,005	134,558
GLM-5.3-Flash	1,385,490	15,494
Kimi-K3	922,396	7,546
Qwen3.8-Max	1,110,492	12,205
Doubao-Seed-2.1-Pro	1,477,011	5,433

Tab. 9: Average token consumption under the ablation settings in Tab. 4.

Model	Setting	Input	Output	Reason.	Steps
GPT-6-ASTRA	Full	2,463,091	9,877	8,582	22.9
	Ask-only	121,413	52,203	50,408	36.5
	Unambig.	2,571,435	4,011	2,545	13.0
Hunyuan-4-Preview	Full	2,138,240	13,298	12,364	20.6
	Ask-only	22,786	19,523	19,058	9.6
	Unambig.	1,844,196	15,872	15,185	11.6
DeepSeek-V4.1-Flash	Full	2,712,005	134,558	133,093	21.3
	Ask-only	17,602	442,020	441,277	10.1
	Unambig.	3,236,902	99,292	97,238	13.1
Doubao-Seed-2.1-Pro	Full	1,477,011	5,433	4,411	12.8
	Ask-only	15,622	14,720	14,397	6.7
	Unambig.	3,720,158	7,362	6,188	9.8

G Reproducibility under a Black-Box Search Backend

All SEARCH calls in DISCOBENCH are routed through Tavily (Tavily Inc., 2026), a hosted web-search API whose index, ranking model, and freshness policies are not publicly disclosed. From the agent's point of view the backend therefore behaves as a black box: the same query issued on different days can return different snippet sets, different rankings, and even different source domains, both because the web itself is non-stationary (pages appear, are edited, or are de-indexed) and because Tavily's own retrieval and re-ranking stack can be updated without notice. Strict bit-exact reproducibility of an end-to-end trajectory is therefore not achievable.

Solvability of individual instances. This stochasticity does not, however, undermine the solvability of individual questions, because the gold answer in DISCOBENCH is by construction time-invariant. Every question is built on stable, verifiable factual knowledge, so the correct answer does not drift as the web changes. Tavily's black-box behavior varies only the *retrieval surface*: which snippet from which source is returned, and in what order. Since the underlying evidence lives in well-established, widely indexed public sources, any reasonably comprehensive web index can be expected to surface at least one supporting snippet for a reasonable query. DISCOBENCH therefore remains solvable in principle on every run.

Variance across runs. The black-box effect manifests not as a change in answerability but as variance in *trajectory shape and retrieval efficiency*: the specific snippets surfaced, their ranking, and therefore which queries are sufficient and how many SEARCH calls an agent needs before the relevant evidence appears in its context window. Two runs of the same agent on the same question may consequently take different paths and consume different numbers of tool calls even when both ultimately succeed.

Implications for interpretation and replication. A model's reported score on DISCOBENCH should be read as an expectation over Tavily snapshots rather than as a per-run deterministic quantity, and an individual incorrect trajectory should be inspected for its underlying cause—retrieval ordering on a given day versus a genuine ambiguity-handling failure—before being attributed to

model capability. To make replications maximally comparable, we recommend that future users of DISCOBENCH (i) run all compared agents within a short, contiguous evaluation window so they observe near-identical Tavily snapshots, and (ii) where feasible, cache and release the raw Tavily responses alongside model outputs, which converts a hard reproducibility problem (re-deriving an identical live web view) into a tractable one (replaying a fixed snippet log).

Trace retention and release. We retain timestamped search logs, query traces, model outputs, and agent trajectories. Subject to provider and data-release permissions, we will release cached retrieval responses with these artifacts so that recorded runs can be audited and replayed against a fixed retrieval surface.

H EXAMPLES

Two cases illustrate the design of DISCOBENCH. Tab. 10 presents Case 1, combining Factual Inaccuracy and Version ambiguity within one retrieval chain. Tab. 11 presents Case 2, where one description admits different ranking criteria. Each lists the original question, the full rewrite as a construction reference, and the checkpoint types, targets, ambiguity logic, and discriminative clues supplied by the user simulator. In every table, English rows (gray tint) precede the original Chinese (cream tint). Colored text tracks each substitution thread: the same color appears on the original constraint, its weakened rewrite, and the clue that later restores it.

Case 1: Factual Inaccuracy followed by Version ambiguity. Tab. 10 presents a four-checkpoint question from the *Sports* domain. Two substitution threads run through the trajectory but follow different injection patterns. Thread A (CP_1) is a *Factual Inaccuracy*: the original country nickname "Land of Windmills" (the Netherlands) is replaced by "Land of Hajimi", a fabricated term that does not refer to any real country. The agent cannot resolve this checkpoint through retrieval alone – a faithful search returns no match – and must invoke ASK rather than guess. Thread B (CP_3) is a *Version* ambiguity: the original match identifier "at the 60th minute of a 2018 CSL match" is weakened to the looser window "in a CSL match in March–April". Within that window Wang Chu came on as a substitute in two different matches on different dates, replacing Wang Gang in one and Cao Yongjing in the other; only the precise date disambiguates which match is meant. The two threads together stress that DISCOBENCH requires the agent to switch *detection mode* within a single trajectory rather than apply a single clarification heuristic uniformly.

Case 2: Criteria ambiguity in a long bridging chain. Tab. 11 presents a three-checkpoint question spanning the *Technology* domain. Unlike the preceding case, here a single colored phrase in the original is *deleted* rather than *replaced*: the qualifier "also nicknamed the 'Ice City'" is removed during ambiguity injection. After deletion, the surviving constraint – "a city whose Chinese name has three characters and lies above 40°N, listed among the top three [beer-festival] cities" – can be satisfied under two distinct enumeration criteria: the world's top three beer festivals (yielding Munich, 慕尼黑, ∼48°N) or China's top three beer festivals (yielding Harbin, 哈尔滨, ∼45°N). The agent must recognize that a single description fits two rankings and clarify *which ranking* the user intends before proceeding.

Tab. 10: **Case 1 (Factual Inaccuracy + Version).** Four-checkpoint question from the *Sports* domain; CP_1 injects a fabricated country nickname, CP_3 injects an under-specified timing window admitting two candidate teammates.

Domain: Sports | **Difficulty:** Medium | **# CPs:** 4 | **# Ambi:** 2 (Factual Inaccuracy, Version) | **Gold:** `Chongqing`

PART 1 — INPUT QUESTION PAIR (original seed and its ambiguity-injected rewrite)

Original Q *(seed, before ambiguity injection; reference only)*

en	A certain country is known as the "Land of Windmills". A city football team is based in this country's seat of government. In 2013, a Chinese player trial-trained with that team; in a 2018 Chinese Super League (CSL) match, this player came on as a substitute for a male athlete at the 60th minute. Where was that male athlete born?
zh	某个国家被称为"风车之国"，该国家的政府所在地有一个城市球队。2013年一名中国球员在该队试训，该球员在2018年中超比赛中第60分钟替补一名男运动员出场，该男运动员出生于哪里?

Rewritten Q *(after ambiguity injection; construction reference)*

en	A certain country is known as the "Land of Hajimi". A city football team is based in this country's seat of government. In 2013, a Chinese player trial-trained with that team; in a CSL match in March–April, this player came on as a substitute for a male athlete. Where was that male athlete born?
zh	某个国家被称为"哈吉米之国"，该国家的政府所在地有一个城市球队。2013年有中国球员在该队试训，该球员在3月～4月中超的某个比赛中替补一名男运动员出场，该男运动员出生于哪里?

PART 2 — CHECKPOINT TRAJECTORY (four sub-questions, evaluated in order)

CP_1 — Ambi *(Factual Inaccuracy)*

Sub-Q	Which country is known as the "Land of Hajimi"? 哪个国家被称为"哈吉米之国"?
Target	the Netherlands 荷兰
Logic	"Land of Hajimi" is not an actual nickname for any country; it is a fabricated reference deliberately injected as a factual inaccuracy. Searches return no plausible match, so the agent must invoke ASK for clarification rather than commit to a guess. "哈吉米之国"并非任何国家的真实别称，是出题阶段刻意杜撰的事实错误。retrieval 不会返回匹配结果，agent 应识别此事实错误并向用户求助，而非凭借猜测作答。
Clue	"I misremembered – it's the Land of Windmills." "我记错了，是风车之国"

CP_2 — Unambi

Sub-Q	A city football team is based in this country's seat of government; in 2013, a Chinese player trial-trained with that team. Who is the player? 该国家的政府所在地有一个城市球队，2013年有位中国球员在该队试训，该球员是?
Target	Wang Chu 王楚

CP_3 — Ambi *(Version)*

Sub-Q	In a CSL match in March–April 2018, this player came on as a substitute for a male athlete. Who is that athlete? 该球员2018年3月～4月中超的某个比赛中替补一名男运动员出场，这个男运动员是?
Target	Cao Yongjing 曹永竞
Logic	Within March–April 2018, Wang Chu came on as a substitute in two distinct CSL matches on different dates, replacing Wang Gang and Cao Yongjing respectively; the rewritten window matches both, and only the exact date pins down which. 王楚在2018年3月～4月期间替补出场两次，在两个不同日期的中超比赛中分别替换了王刚和曹永竞；重写后的时间窗口同时匹配这两场，唯有具体日期才能锁定其中一场。
Clue	"It was on April 8, 2018." "是在2018年4月8日上场的"

CP_4 — Unambi *(final)*

Sub-Q	Where was this athlete born? 这个运动员出生于哪里?

Target: `Chongqing` / 重庆市

Tab. 11: **Case 2 (Criteria).** Three-checkpoint question in the *Technology* domain; one Criteria-type Ambi checkpoint (CP_2). The substitution thread here is *removed* rather than replaced, so the colored phrase appears only in the original question and in the discriminative clue.

Domain: Technology | **Difficulty:** Easy | **# CPs:** 3 | **# Ambi:** 1 (Criteria) | **Gold:** 12

PART 1 — INPUT QUESTION PAIR (original seed and its ambiguity-injected rewrite)

Original Q *(seed, before ambiguity injection; reference only)*

en	The R&D of a certain manned submersible was designated as a key special project under the National 863 Program in 2002. The submersible made its first dives in the world's third-largest ocean in a certain year; in April of that same year, a research vessel completed its maiden voyage from a certain dock. That dock is a sub-project of a certain plaza, which opens to visitors free of charge during a certain folk festival. The festival is associated with a "top three [cities]" ranking; among the three cities, one has a Chinese name consisting of three characters and lies above 40°N and is also nicknamed the "Ice City". A satellite launched from the Jiuquan Satellite Launch Center on 9 June 2023 was jointly developed by a university located in this city and a company. According to the university's official faculty overview updated on 28 November 2025, how many Ministry of Education science and technology innovation teams does it have?
zh	某载人潜水器的研制工作在2002年被列为863计划重大专项，该载人潜水器在某年首次奔赴世界第三大洋开展下潜作业，同年4月，某考察船在某码头完成首次启航，该码头是某广场的子项目，该广场在某一节日期间免费向游人开放，该民间节日有一个榜单，榜单的三大城市中有一个三字城市位于北纬40多度且别称是"冰城"，某颗于2023年6月9日在酒泉卫星发射中心发射的卫星，由位于该城市的一所大学与一家企业联合研制。根据该大学于2025年11月28日更新的官方师资队伍介绍，其教育部科技创新团队有多少个?

Rewritten Q *(after ambiguity injection; construction reference)*

en	Identical to the Original Q above, with the colored fragment "and is also nicknamed the 'Ice City' " *deleted*.
zh	与上方原始问句完全一致，仅删除标红片段"且别称是'冰城' "。

PART 2 — CHECKPOINT TRAJECTORY (three sub-questions, evaluated in order)

CP_1 — Unambi

Sub-Q	(Chained from the early constraints:) the dock is a sub-project of a plaza that opens free of charge during which folk festival?
	某载人潜水器的研制工作在2002年被列为863计划重大专项，该载人潜水器在某年首次奔赴世界第三大洋开展下潜作业，同年4月，某考察船在某码头完成首次启航，该码头是某广场的子项目，该广场在某一节日期间免费向游人开放，这个节日是?
Target	the Beer Festival (啤酒节)
	啤酒节

CP_2 — **Ambi** *(Criteria)*

Sub-Q	The folk festival is associated with a "top three [cities]" ranking; one of the three cities has a Chinese name of three characters and lies above 40°N. Which ranking is being referred to?
	该民间节日有一个榜单，榜单中有一个三字城市位于北纬40多度，这个榜单是什么?
Target	China's Top Three Beer Festivals
	中国三大啤酒节
Logic	"Top Three Beer Festivals" admits two distinct enumeration criteria: *the world's* top three versus *China's* top three. Each list contains a city whose Chinese name has three characters and lies above 40°N — Munich (慕尼黑, ∼48°N) in the world list, Harbin (哈尔滨, ∼45°N) in the China list. The rewritten constraint alone cannot tell the two lists apart.
	"三大啤酒节"这一榜单的口径存在二义性：可指"世界三大啤酒节"或"中国三大啤酒节"。两份榜单各自包含一座中文名为三字、纬度在北纬40°以上的城市——世界榜中的慕尼黑（约48°N）与中国榜中的哈尔滨（约45°N）。重写后的约束本身无法区分这两份榜单。
Clue	"The three-character city is also nicknamed the 'Ice City'."
	"三字城市别称是'冰城'"

CP_3 — Unambi *(final)*

Sub-Q	A satellite launched from the Jiuquan Satellite Launch Center on 9 June 2023 was jointly developed by a university located in Harbin and a company. According to the university's official faculty overview updated on 28 November 2025, how many Ministry of Education science and technology innovation teams does it have?
	某颗于2023年6月9日在酒泉卫星发射中心发射的卫星，由位于哈尔滨的一所大学与一家企业联合研制。根据该大学于2025年11月28日更新的官方师资队伍介绍，其教育部科技创新团队有多少个?

Target: 12

Example sources and correction. This displayed example corrects the launch location, team category, and reference date. Satellite details follow the university report of 9 June 2023; the team count follows the official faculty overview updated on 28 November 2025.

I ANNOTATION DETAILS

Recruitment and compensation. Annotators and quality inspectors were undergraduate students recruited from multiple institutions, with diverse academic backgrounds across several disciplines. They were compensated on a per-item (piece-rate) basis, with a total payout of $39,000 for the entire annotation effort.

Annotator consent. All annotators and inspectors were informed in advance that their annotations would be released as part of a public benchmark and consented to this use.

Ethics review. The annotation task involved creating factual question–answer pairs from publicly available web resources and did not involve the collection of personal or sensitive information, so IRB approval was not required.

J Quality Inspection

This section provides further details on the quality control (QC) process introduced in Section 4.3. During the two-phase construction pipeline (Section 4), the initial LLM-assisted generation and human annotation produced a larger pool of candidate samples. After preliminary filtering for deduplication, format compliance, and basic factual verification, 314 candidate samples were retained. We then applied a multi-stage QC pipeline to these 314 samples to identify and remove low-quality items before assembling the final benchmark. The pipeline combines automatic structural checks, LLM-based probing, and manual review, and employs a multi-agent architecture in which a coordinating agent dispatches candidate samples in batches to specialized sub-agents operating under strictly constrained prompts.

Stage 1: Structural Validation. An automatic script verifies every candidate sample for field completeness (all required fields non-empty), checkpoint-structure consistency (each sample contains at least one *Ambi* and one terminal checkpoint; every *Ambi* node carries a non-empty `ambiguity_logic` and `clue_if_asked`), and difficulty–label alignment (Easy / Medium / Hard corresponds to 1 / 2 / 3 ambiguity checkpoints, respectively).

Stage 2: LLM-Based Probing. Each sample is independently tested under two complementary conditions to screen for potential knowledge shortcuts and clarification-free solvability. In both cases, the sub-agent receives the *complete rewritten question* (i.e., the full user query after ambiguity injection) rather than individual checkpoint sub-questions. The prompt templates are provided in Box J.

- **Closed-book probing.** The sub-agent answers the full question using only parametric knowledge, with all retrieval tools disabled. A correct answer signals potential *knowledge leakage*. Because the complete question may expose intermediate entities that would not be visible when checkpoints are processed sequentially, flagged samples are individually reviewed to distinguish genuine leakage from artifacts of the holistic testing format.
- **Open-book probing.** The sub-agent is given access to a search tool (capped at 25 calls to prevent runaway retrieval loops) but is strictly prohibited from asking clarification questions. A correct answer under this constraint signals *clarification-free solvability*: the injected ambiguity may not effectively require multi-turn clarification.

Answer equivalence between sub-agent outputs and ground-truth answers is determined by a separate LLM-based judge, accounting for surface-form variations such as transliterations, date formats, and title markers.

Scope of the probes. Full-question probing screens candidate items during construction. Probes use the complete rewrite, whereas checkpoint labels refer to sequentially disclosed information; manual review interprets probe outcomes in light of these information conditions. Filtering surface-level ambiguity focuses the retained samples on clarification needs exposed through retrieval.

Stage 3: Ambiguity and Factual-Error Assessment. For each ambiguous checkpoint, a sub-agent assesses whether the ambiguity is *surface-level*, i.e., whether a typical user could enumerate the candidate entities from the question text alone using only commonsense knowledge. Surface-level ambiguity suggests that an agent could resolve the checkpoint by simply asking the user to choose among obvious candidates, without performing any retrieval. Separately, for checkpoints of the *Factual Inaccuracy* type, the sub-agent evaluates whether the injected error is recognizable without retrieval. Errors that a typical user could identify through commonsense alone (e.g., historically impossible dates or well-known factual contradictions) undermine the intended interaction pattern, as the agent should need retrieval evidence to detect and challenge such inaccuracies.

Stage 4: Manual Review and Answer Verification. All automatically flagged samples undergo manual review covering three aspects: (1) *question and ambiguity design*, including whether the question text uniquely constrains the expected answer, whether the injected ambiguity is realistically

triggerable during retrieval, and whether sub-questions are logically coherent with the overall reasoning chain; (2) *clue and retrieval quality*, including whether the discriminative clue is natural and sufficient for disambiguation, and whether the target answer is retrievable through mainstream search engines; and (3) *answer correctness*, where we cross-check ground-truth annotations against external sources. When the open-book sub-agent produces a plausible alternative answer differing from the ground truth, we verify whether the discrepancy reflects a legitimate alternative interpretation or an annotation error, and correct or supplement the ground truth where necessary. Samples with correctable issues are revised; only samples with fundamental design flaws are removed.

Stage 5: Final Verdict. A rule-based decision tree aggregates the signals from the preceding stages to screen samples for the intended retrieval-dependent clarification setting. A sample is removed when its quality signals indicate otherwise: commonsense-recognizable factual errors are removed because they do not require retrieval to detect; knowledge leakage combined with clarification-free solvability is removed because neither retrieval nor interaction is necessary; and surface-level ambiguity combined with clarification-free solvability is removed because the disambiguation does not depend on retrieved evidence. Individual weak signals that may stem from the holistic prompt format are not grounds for removal on their own, but are noted for inspection.

Results. Tab. 12 summarizes the QC outcomes. Of the 314 candidate samples, 236 (75.2%) passed quality control and 78 (24.8%) were removed. The final DISCOBENCH benchmark comprises 211 samples drawn from those that passed, forming the common subset evaluable across all tested models after accounting for content-policy restrictions of individual model providers.

QC PROBING PROMPTS

```
– Closed-book probing –
You are a knowledge assistant. Answer the following question using only your training
data.
[Mandatory rules]
1. You must NOT use any tools (WebSearch, WebFetch, Bash, Read, etc.).
2. You must NOT say "I need to search" or "I cannot determine." Even if uncertain,
give your best guess.
3. Do not explain your reasoning. Output the answer directly.
Question: {question}
Output exactly one line of JSON:
{"task_id":"<id>", "answer":"<your answer>",
  "confidence":"high|medium|low",
  "used_tools":false}
– Open-book probing –
You are a research assistant. You may use WebSearch and WebFetch tools.
[Mandatory rules]
1. You may use WebSearch and WebFetch.
2. You must NOT ask the user for clarification or say "the question is ambiguous."
Even if ambiguous, make your best judgment based on retrieval results and give a
final answer.
3. Total WebSearch and WebFetch calls must not exceed 25. After 25 calls, give your
best guess based on available information.
4. You may only answer once; do not split across multiple turns.
Question: {question}
Output exactly one line of JSON:
{"task_id":"<id>", "answer":"<your answer>",
  "search_count":<int>,
  "ambiguity_noticed":true|false}
```

K EVALUATION HARNESS VALIDATION

We validate the responder against independent human judgments at the final-answer and checkpoint levels. For final answers, human correctness judgments serve as the reference. For checkpoints, annotators inspect the benchmark ground truth and agent trace to determine whether the responder decision is correct.

The final-answer responder agrees closely with human references. At the checkpoint level, human auditors accept 457 of 459 sampled simulator/judge decisions (99.6%).

Tab. 12: Quality control results on the 314 candidate samples.

Category	**N**	**%**
Overall QC outcome		
Passed	236	75.2
Removed	78	24.8
Removal reasons (N = 78)		
Commonsense factual error	49	62.8
Leakage + solvable w/o clarif.	21	26.9
Knowledge leakage only	3	3.8
Solvable w/o clarif. + surface amb.	3	3.8
Other (structural / answer defects)	2	2.6

Tab. 13: Final-answer judge validation against human references.

Reference	**n**	**Agr.**	κ	**F1**
Individual annotations	139	98.6	96.6	97.6
Majority vote	79	97.5	94.0	95.8

We also evaluate the quality of simulator replies with four annotators on a three-point scale; non-applicable cases are excluded from the corresponding dimension.

Tab. 14: Human evaluation of simulator replies.

Dimension	**Scored n**	**Mean**	**Raw agr.**
Relevance	66	2.970	0.842
Discriminativeness	76	2.961	0.608
Naturalness	66	2.924	0.808
Non-leakage	79	3.000	0.658

All applicable dimensions score at least 2.924/3, and no scorable reply is judged to leak the final answer. For independent checkpoint-trajectory annotation, the 244 checkpoints completed by all four annotators obtain 98.4% average pairwise raw agreement and Gwet's AC1 of 0.983.

The audits assess judgment and reply consistency against the annotated checkpoint states and clue rules.

L Responder Prompt Robustness

We test two variants of the responder prompt. V1 rephrases presentation style and V2 reorganizes the prompt; both retain the same input fields, decision matrix, hard constraints, output schema, and state-transition rules as V0. For fixed-input analysis, each version receives the same recorded checkpoint configuration, dialogue history, and agent action; only the responder prompt changes.

Tab. 15: Fixed-input agreement across responder-prompt versions (%).

Comparison	**Status**	**Clue**	**Gold clue**	**Final state**
V0 vs. V1	91.8	95.2	89.6	98.3
V0 vs. V2	90.4	97.3	89.6	97.4

Status denotes exact structured-status agreement; Clue denotes agreement on whether to release a clue; Gold clue denotes semantic fidelity to the reference clue when a clue is required; Final state denotes agreement on the terminal decision. These results show stable clue-release and final-state decisions under prompt rephrasing, while retaining some decision-level variation.

The end-to-end ranges are 0.6–1.3 points and have no uniform direction across models. We therefore interpret close scores as performance tiers under a fixed disclosed protocol, rather than prompt-invariant fine-grained rankings.

Tab. 16: End-to-end accuracy under responder-prompt variants (%).

Model	V0	V1	V2	Range
GPT-6-ASTRA	63.7	63.7	62.8	0.9
Gemini-3.8-Flash	46.1	46.3	46.7	0.6
Doubao-Seed-2.1-Pro	28.4	27.6	28.9	1.3

M STATISTICAL ROBUSTNESS

We assess sensitivity to the question composition with 1,000 question-level bootstrap resamples and uniform reweighting of the four ambiguity types by their primary ambiguity point. Reweighted rankings are highly consistent with the original ranking (Spearman $\rho = 0.991$; Kendall $\tau = 0.964$). Across bootstrap resamples, mean Spearman correlation with the original ranking is 0.957, and the Top-3 set is unchanged in 84.2% of resamples. These analyses support stable overall comparisons, while not supporting a definitive ordering of closely scored models. Resampling complete questions preserves their trajectory structure to assess overall ranking sensitivity to question composition. Reweighting by primary ambiguity type likewise tests question-level composition; checkpoint metrics retain their reached-state populations and the conditional interpretation defined in Appendix A.

Results in Tab. 2 are means over three runs on the same 211 questions. Across-run standard deviations in end-to-end accuracy are 0.60, 0.85, and 1.25 percentage points for GPT-6-ASTRA, DeepSeek-V4.1-Flash, and Doubao-Seed-2.1-Pro, respectively.

Scale and evaluation cost. DISCOBENCH is an interaction-intensive agent benchmark rather than a static QA set. Each instance involves multi-step search, checkpoint decisions, user clarification, simulator interaction, and LLM-based evaluation, making construction and evaluation substantially more costly per item than ordinary answer-only QA. Its scale is comparable to recent interactive agent benchmarks, such as InteractComp (210 instances) (Deng et al., 2026), AssistantBench (214 instances) (Yoran et al., 2024), and IDRBench (100 instances) (Feng et al., 2026). A full run over all evaluated models costs approximately $7,877; at within-model concurrency of 10, single-model runs take 5h09m–14h39m, totaling 105h48m across models.

N PROMPT TEMPLATES

N.1 MULTI-TURN RESPONDER PROMPT

MULTI_TURN_RESPONDER_PROMPT

```
Role: Multi-turn dialogue simulated user
You play the role of a real user who is having a multi-turn dialogue with an
intelligent assistant (Agent), guiding it step by step through checkpoints to
complete a complex query task.
Your responsibility: based on the current-turn configuration and the Agent's
utterance in this turn, determine the state and output exactly one valid JSON object.
1. Task Configuration
Final answer (final_answer): {final_answer}
Forbidden information (forbidden_info) - These strings must never appear directly in
reply; they may only be hinted at indirectly through clue_if_asked:
{forbidden_info}
Note: You can only see the information under "Current Turn" below. You cannot see
fields from any other checkpoint. This is a strict guarantee of progressive
disclosure. Make judgments only within the context of the current step.
2. Current Turn State
- Current step: Step {current_step} / {total_steps}
- Whether this is the last step: {is_last_checkpoint} (values: yes / no; treat it as
the last step only when it equals yes)
- Standard question for this step, question (the system has automatically appended it
to the Agent after the previous reply): {current_question}
- Goal for this step, goal: {current_goal}
- Ambiguity description for this step, ambiguity_logic: {current_ambiguity_logic}
- Clue for this step, clue_if_asked: {current_clue_if_asked}
```

```
(If it is null or empty, this step has no ambiguity; otherwise, this step has
ambiguity.)
- Whether the ambiguity in this step has been clarified, ambiguity_clarified:
{current_ambiguity_clarified} (values: yes / no)
- This is maintained by the system state machine. It changes to yes when the previous
status was ambiguity_detected, and resets to no when entering a new checkpoint.
- This field is authoritative: if it is yes, it means the Agent has already asked
about this step's ambiguity and received the clue. It is forbidden to classify the
turn as any missed_ambiguity status.
- If it is no, this step has ambiguity, and the Agent directly uses answer, then use
case 3c or 3w.
- Agent tool hint for this turn, tool_name_hint: {tool_name_hint} (values: ask /
answer / unknown)
3. Global Hard Constraints (check each item before output)
- H1. reply must not contain any forbidden_info string. Do not repeat entity names
even in an affirmative reply.
- H2. reply must not repeat or imply the next question. The system will automatically
append the next checkpoint's standard question after your reply; repeating it would
cause the Agent to see two nearly identical user messages.
- H3. reply should use everyday spoken language ("yes", "right", "found it") and must
not contain meta terms such as "checkpoint", "configuration", "goal", or
"ambiguity_logic".
- H4. Progressive disclosure: provide feedback only for the current step, and do not
reveal information from later steps.
- H5. The output must be valid JSON, with no Markdown code block or any text outside
the JSON object.
4. Decision Matrix (match from top to bottom; stop at the first match)
1c. Trigger: tool_name_hint == ask and the Agent asks exactly about the ambiguity
point corresponding to this step's ambiguity_logic. Status: ambiguity_detected.
1w. Trigger: tool_name_hint == ask, this step has ambiguity, and the Agent asks about
another ambiguity, a later step, or an irrelevant clarification. Status:
wrong_ambiguity_detected.
5c. Trigger: this is the last step (is_last_checkpoint == yes) and the Agent's final
answer matches final_answer. Status: task_completed.
5w. Trigger: this is the last step and the final answer does not match final_answer.
Status: wrong_answer.
2c. Trigger: tool_name_hint == answer, the answer contains the key entity in goal and
is logically correct, and either this step has no ambiguity or ambiguity_clarified ==
yes. Status: correct_answer.
2w. Trigger: tool_name_hint == answer, this step has no ambiguity, and the answer
does not contain the key entity or is logically wrong. Status: wrong_answer.
3c. Trigger: tool_name_hint == answer, this step has ambiguity, ambiguity_clarified
== no, and the answer happens to be correct. Status: missed_ambiguity_correct.
3w. Trigger: tool_name_hint == answer, this step has ambiguity, ambiguity_clarified
== no, and the answer is wrong. Status: missed_ambiguity_wrong.
4. Trigger: tool_name_hint == unknown, and agent_input is neither a question nor an
answer statement (pure reasoning, search only, or empty text). Status: search_only.
4.1 Conflict Between Hint and Text
- If tool_name_hint == ask but agent_input is clearly a declarative answer, treat it
as tool misuse and handle it as case 2w or 3w. Set failure_reason to tool_misuse:
<one sentence>.
- If tool_name_hint == answer but agent_input is clearly a question, also treat it as
tool misuse and handle it as case 1c or 1w.
4.2 Basis for Determining Historical Clarification
Primary source: directly inspect the ambiguity_clarified field in Section 2. If it is
yes, the ambiguity has been clarified, and it is forbidden to classify the turn as
any missed_ambiguity status.
Supplementary source: the end of the dialogue history may include [Clarification
records already occurred in this checkpoint] (appended by the system from an
authoritative data source; absolutely reliable and unaffected by truncation). This is
only a human-readable note. If it conflicts with ambiguity_clarified, follow
ambiguity_clarified.
5. Replies and JSON Templates for Each Case
```

```
Case 1c: ambiguity_detected
- If clue_if_asked is non-empty, state it in natural spoken language.
- If clue_if_asked is empty, write: "There should be no ambiguity in this step; you
can continue searching."
Example (Agent: "There are two candidate works in that year. Which one do you mean?"):
{"status": "ambiguity_detected", "should_continue": true, "failure_reason": null,
"reply": "I remember it was an urban sci-fi TV series."}
Case 1w: wrong_ambiguity_detected
- Do not reveal clue_if_asked, and do not mention any forbidden_info.
- Decline naturally in spoken language.
{"status": "wrong_ambiguity_detected", "should_continue": true, "failure_reason":
"wrong_ambiguity_detected", "reply": "That is not the point I need to clarify."}
Case 2c: correct_answer (not the last step)
- reply should be only a short confirmation, without entity names and without
repeating the next question.
- Acceptable replies: yes / right / found it / that's it / mhm.
{"status": "correct_answer", "should_continue": true, "failure_reason": null,
"reply": "yes"}
Case 2w: wrong_answer (not the last step, no ambiguity)
- reply must be exactly wrong.
- Set failure_reason to wrong_answer: <one-sentence reason>.
{"status": "wrong_answer", "should_continue": false, "failure_reason": "wrong_answer:
the returned entity does not match the correct answer", "reply": "wrong"}
Case 3c: missed_ambiguity_correct
- reply should be a short confirmation, without entity names and without repeating the
next question.
{"status": "missed_ambiguity_correct", "should_continue": true, "failure_reason":
"missed_ambiguity: the Agent directly gave the correct answer without asking which
candidate was intended", "reply": "right"}
Case 3w: missed_ambiguity_wrong
- reply must be exactly wrong.
{"status": "missed_ambiguity_wrong", "should_continue": false, "failure_reason":
"missed_ambiguity: the Agent gave a wrong answer without asking about the ambiguity",
"reply": "wrong"}
Case 4: search_only
- The Agent used neither ask nor answer in this turn (search only, pure reasoning, or
empty text).
- Write a light prompt such as Did you find anything? / Hmm?.
{"status": "search_only", "should_continue": true, "failure_reason": "search_only",
"reply": "Did you find anything?"}
Case 5c: task_completed (last step and final answer is correct)
- reply must be exactly correct; no other content is allowed.
{"status": "task_completed", "should_continue": false, "failure_reason": null,
"reply": "correct"}
Case 5w: wrong_answer (last step and final answer is wrong)
- reply must be exactly wrong.
{"status": "wrong_answer", "should_continue": false, "failure_reason": "wrong_answer:
final answer does not match final_answer", "reply": "wrong"}
6. Output Schema (single authoritative schema)
{
"status": "ambiguity_detected | wrong_ambiguity_detected | correct_answer |
wrong_answer | missed_ambiguity_correct | missed_ambiguity_wrong | search_only |
task_completed",
"should_continue": true,
"failure_reason": "null | <tag> | <tag>: <one sentence>",
"reply": "one spoken English sentence shown to the Agent"
}
- status: required; must be one of the enum values above.
- should_continue: required Boolean; cases 2w / 3w / 5c / 5w are false, all others are
true.
- failure_reason: successful cases must use null; for failures, prefer the following
tags (optionally append : <one sentence>):
- wrong_answer, missed_ambiguity, wrong_ambiguity_detected, search_only, tool_misuse
```

```
- reply: required, and must satisfy all H1-H4 constraints in Section 3.
7. Dialogue History
{conversation_history}
8. Agent Input for This Turn
{agent_input}
9. Your Output
Output exactly one JSON object according to the schema in Section 6. Do not output
any other text.
Before output, check mentally (do not write this into reply):
- reply contains no forbidden_info
- reply does not repeat the next question
- reply uses spoken language
- status, should_continue, and failure_reason are mutually consistent
- The entire output is valid JSON and contains exactly one JSON object
```

N.2 NEUTRAL SYSTEM PROMPT

Neutral denotes the base prompt without the additional ambiguity emphasis of Guided, not an unconstrained policy: it requests retrieval before choosing further search, clarification, or an answer. The prompt text below is retained as used.

NEUTRAL_SYSTEM_PROMPT

```
Task Description
You are an intelligent agent having a multi-turn dialogue with the user to complete a
complex query task step by step. Each step is a complete question. You need to search
for information and provide an answer.
Multi-turn Interaction Mode
This is a multi-turn dialogue task. The user will provide sub-questions step by step.
For each sub-question, follow these requirements:
1. Use the search tool to search for relevant information. Multiple search calls may
be issued in parallel within one step.
2. Based on the search results obtained, perform one of the following actions:
1. If the information after searching is insufficient to uniquely identify the
answer, call the ask tool to request clarification.
2. If the information is insufficient but can be resolved through additional
retrieval, continue searching and return to step 2.
3. If the search results are clear, call the answer tool to provide the final answer
for this step.
Special notes:
1. Strictly distinguish between the ask and answer tools.
2. Every turn must end with a call to either ask or answer. Once you have obtained
the answer or need to interact with the user, you must end the search phase by
calling either ask or answer.
Tool Responsibilities
Tool 0: search - retrieve online information
Calling rules:
- Use it to collect facts related to the current question from online information
sources.
- Multiple different queries may be issued in parallel within one turn, and the
results will be returned together.
- Queries should use precise keyword combinations (entity name + limiting dimension,
e.g., "Qi Wei 2021 TV drama urban sci-fi") and avoid full-sentence natural language
queries.
- Under a single question, historical query results remain in the context and can be
inspected at any time.
- If Chinese entities are involved (person names / film and television / companies /
place names), Baidu Baike or Chinese-site keywords may be prioritized.
Tool 1: ask - ask the user a clarification question
Calling rules:
- When the information required for the current sub-question cannot be obtained
through retrieval alone, request an additional clue from the user.
```

- The question should focus on the specific missing discriminative dimension, using comparative or directional wording so that the user can reply easily.
- Ask about only one dimension at a time so that the returned clue is singular and usable.
- Do not ask the user for facts that can be obtained through search.

Tool 2: answer – submit the final answer
Calling rules:
- Call it when the answer to the sub-question can be determined from the obtained information.
- Once called, the final answer is submitted, and no further ask or search operations can be performed under the current sub-question.

Important Constraints
- Every turn must end with a tool call; plain text without a tool call is not accepted.
- Do not misuse tools: when interaction with the user is needed, use ask, not answer; when a clear result should be returned, use answer, not ask.
- Within the same turn, ask / answer may be called at most once. Once ask or answer is called, the current turn ends immediately, and the next turn can begin only after the user replies.
- Under each sub-question, you can see all search results and ask / answer interaction logs. Reuse existing information first to avoid duplicate searches.

Answer Format Requirements (only for the answer field of the answer tool)
When giving an answer (not when actively interacting with the user), the answer should:
- Be as concise as possible and directly answer the user's question.
- Not contain explanations or narrative descriptions.
- Use Arabic numerals if it is a number.
- Not use articles or abbreviations if it is a string.

Before calling any tool, think step by step: Is the current information sufficient to uniquely identify the answer? If yes, use answer; if no, continue search or use ask to interact with the user.

N.3 GUIDED SYSTEM PROMPT

GUIDED_SYSTEM_PROMPT

Task Description
You are an intelligent agent having a multi-turn dialogue with the user to complete a complex query task step by step. Each step is a complete question. You need to search for information and provide an answer.

Question Prior (Important)
The questions in this task usually deliberately replace proper names with "shared features / vague references". The question text may point to multiple candidate entities at the same time. The discriminative facts needed to distinguish candidates are held by the user and will be released only if you proactively use ask. Therefore:
- If literal searching returns two or more candidates that all satisfy the constraints, this is not a retrieval failure but the intended form of the question. The correct action is ask, not choosing one candidate and using answer.
- Do not make a guessed answer based on which candidate is more popular or more likely. Even if the guess is correct, it still counts as missing the ambiguity.
- A small number of non-ambiguous bridging steps may also exist, where the search result is unique. In those cases, directly use answer. The criterion is the number of search-result candidates, not the length of the question.

Multi-turn Interaction Mode
This is a multi-turn dialogue task. The user will provide sub-questions step by step. For each sub-question, follow these requirements:
1. Use the search tool to search for relevant information. Multiple search calls may be issued in parallel within one step.
2. Based on the search results obtained, perform one of the following actions:
1. If searching finds multiple candidates or the information is insufficient to uniquely identify the answer, call the ask tool to request clarification from the user.
2. If the information is insufficient but can be resolved through additional retrieval, continue searching and return to step 2.

3. If the search results clearly identify a single entity, call the answer tool to provide the final answer for this step.
Special notes:
1. Strictly distinguish between the ask and answer tools.
2. Every turn must end with a call to either ask or answer. Once you have obtained the answer or need to interact with the user, you must end the search phase by calling either ask or answer.
Tool Responsibilities
Tool 0: search – retrieve online information
Calling rules:
- Use it to collect facts related to the current question from online information sources.
- Multiple different queries may be issued in parallel within one turn, and the results will be returned together.
- Queries should use precise keyword combinations (entity name + limiting dimension, e.g., "Qi Wei 2021 TV drama urban sci-fi") and avoid full-sentence natural language queries.
- Under a single question, historical query results remain in the context and can be inspected at any time.
- If Chinese entities are involved (person names / film and television / companies / place names), Baidu Baike or Chinese-site keywords may be prioritized.
Tool 1: ask – ask the user a clarification question
Calling rules:
- When searching returns two or more candidates that all satisfy the question constraints, or when the required information cannot be obtained through retrieval alone, request an additional clue from the user.
- The question should focus on the specific missing discriminative dimension, using comparative or directional wording (e.g., "Is it A or B?" / "Which feature is it closer to?").
- Ask about only one dimension at a time so that the returned clue is singular and usable.
- Do not ask the user for facts that can be obtained through search.
Tool 2: answer – submit the final answer
Calling rules:
- Call it only after a single candidate entity has been uniquely identified and the current question can be answered directly.
- Once called, the final answer is submitted, and no further ask or search operations can be performed under the current sub-question.
Important Constraints
- Every turn must end with a tool call; plain text without a tool call is not accepted.
- Do not misuse tools: when search results contain multiple candidates, use ask; do not submit a guess with answer. When a unique candidate is identified, use answer, not ask.
- Within the same turn, ask / answer may be called at most once. Once ask or answer is called, the current turn ends immediately, and the next turn can begin only after the user replies.
- Under each sub-question, you can see all search results and ask / answer interaction logs. Reuse existing information first to avoid duplicate searches.
Answer Format Requirements (only for the answer field of the answer tool)
When giving an answer (not when actively interacting with the user), the answer should:
- Be as concise as possible and directly answer the user's question.
- Not contain explanations or narrative descriptions.
- Use Arabic numerals if it is a number.
- Not use articles or abbreviations if it is a string.
Before calling any tool, think step by step: Have the current search results uniquely identified one entity? If yes, use answer; if multiple candidates are found or information is insufficient, continue search or use ask to interact with the user.

N.4 SYSTEM PROMPT WITHOUT SEARCH

The prompt below is the recorded Ask-only configuration in Tab. 4. It disables SEARCH, directs fact verification to ASK, and prohibits guesses from memory, common sense, or probability. The reported results characterize this complete restricted interaction-only configuration.

SYSTEM_PROMPT_w/oSEARCH

```
Task Description
You are an intelligent agent having a multi-turn dialogue with the user to complete a
complex query task step by step. Each step is a complete question. You need to obtain
the necessary information and provide the answer using only multi-turn interaction
with the user.
Environment Constraint (Important)
This task provides no retrieval / search tool. You cannot access the Internet or any
external knowledge base.
- Any detail that requires fact verification must be obtained by using ask.
- You are not allowed to make a guessed answer based on memory, common sense, or
probability. Even if the guess is correct, it still counts as missing the ambiguity.
- The only two tools are ask and answer.
Multi-turn Interaction Mode
This is a multi-turn dialogue task. The user will provide sub-questions step by step.
For each sub-question, follow these requirements:
1. Read the current sub-question and identify discriminative points that require
clarification, such as vague references, multiple candidates, or missing dimensions.
2. Based on the information already available, perform one of the following actions:
1. If the information is insufficient to uniquely identify the answer, call the ask
tool to request clarification from the user.
2. If the information is sufficient to uniquely identify the answer, call the answer
tool to provide the final answer for this step.
Special notes:
1. Strictly distinguish between the ask and answer tools.
2. Every turn must end with a call to either ask or answer. Once you have obtained
the answer or need to interact with the user, you must end the current turn by
calling either ask or answer.
Tool Responsibilities
Tool 0: ask - ask the user a clarification question
Calling rules:
- When the information required for the current sub-question cannot be determined
from the existing context alone, request an additional clue from the user.
- The question should focus on the specific missing discriminative dimension, using
comparative or directional wording (e.g., "Is it A or B?" / "Which feature is it
closer to?").
- Ask about only one dimension at a time so that the returned clue is singular and
usable.
- Because there is no search, any factual detail (person name / time / value / ranking
/ affiliation, etc.) should be asked about whenever it is not yet certain. Do not
assume it yourself.
Tool 1: answer - submit the final answer
Calling rules:
- Call it only after a single candidate entity has been uniquely identified and the
current question can be answered directly.
- Once called, the final answer is submitted, and no further ask operations can be
performed under the current sub-question.
Important Constraints
- Every turn must end with a tool call; plain text without a tool call is not accepted.
- Do not misuse tools: when information is insufficient, use ask; do not submit a
guess with answer. When a unique candidate is identified, use answer, not ask.
- Within the same turn, ask / answer may be called at most once. Once ask or answer is
called, the current turn ends immediately, and the next turn can begin only after the
user replies.
- Under each sub-question, you can see all ask / answer interaction logs. Reuse
existing clues first.
Answer Format Requirements (only for the answer field of the answer tool)
```

```
When giving an answer (not when actively interacting with the user), the answer
should:
- Be as concise as possible and directly answer the user's question.
- Not contain explanations or narrative descriptions.
- Use Arabic numerals if it is a number.
- Not use articles or abbreviations if it is a string.
Before calling any tool, think step by step: Is the currently known information
sufficient to uniquely identify the answer? If yes, use answer; if no, use ask to
request the key clue from the user.
```

N.5 System Prompt without Ask

No-Ask in Tab. 4 retains search and answer actions while disabling questions to the user; Full allows all three actions. Their task-accuracy difference describes performance under these two interaction configurations. Ask-only disables search and directs fact verification to user interaction; Unambiguous provides a reference with target constraints supplied. Neutral–Guided compares prompting policies with the same available actions. Behavioral profiles summarize outcomes after model-selected actions and complement these configuration-level comparisons.

SYSTEM_PROMPT_w/oASK

```
Task Description
You are an intelligent agent. You need to solve a query question through search and
provide the correct answer.
Tool Responsibilities
Tool 0: search - retrieve online information
Calling rules:
- Use it to collect facts related to the current question from online information
sources.
- Multiple different queries may be issued in parallel within one turn, and the
results will be returned together.
- Queries should use precise keyword combinations (entity name + limiting dimension)
and avoid full-sentence natural language queries.
- If Chinese entities are involved (person names / film and television / companies /
place names), prioritize Baidu Baike or Chinese-site keywords.
- Avoid arbitrary URL navigation. Do not directly visit specific URLs that may not
exist; obtain relevant URLs through a search engine.
Tool 1: answer - submit the final answer
Calling rules:
- Call it when the answer can be determined from the search results.
- Once called, the final answer is submitted and cannot be retracted.
Important Constraints
- Every turn must end with a tool call; plain text without a tool call is not accepted.
- Seeking help from a human is strictly forbidden.
- Results such as "failure", "I cannot answer", or "not found" are not accepted. You
must keep searching until the answer is found.
- Your answer must strictly follow the output format required by the task
(alphabetical order, ordering, units, rounding rules, number of decimal places,
etc.).
Answer Format Requirements (constraining the answer parameter of the answer tool)
- It should be a number, or an as-short-as-possible phrase, or a comma-separated list
consisting of numbers and/or strings.
- Do not include explanations or narrative descriptions of the answer.
- Numbers: use Arabic numerals. Do not use thousands separators, and unless otherwise
specified, do not include unit symbols such as $ or %.
- Strings: do not use articles or abbreviations (e.g., for city names).
- Comma-separated lists: apply the corresponding rules above depending on whether
each list element is a number or a string.
Before calling the answer tool, think step by step: Is the current information
sufficient to uniquely identify the answer? If yes, call answer; if no, continue
search.
```