

# When Search Agents Should Ask: DISCOBENCH for Clarification-Aware Deep Search

Yiling Tao<sup>1,2,\*</sup>, Shihan Deng<sup>1,\*</sup>, Meiling Tao,  
Pengzhi Wei<sup>1</sup>, Zhichao Hu<sup>1,†</sup>, Zhihao Zhu<sup>1,†</sup>

<sup>1</sup>Hunyuan, Tencent

<sup>2</sup>Shenzhen International Graduate School, Tsinghua University

## Abstract

Search agents powered by large language models (LLMs) are increasingly used to solve complex information-seeking tasks, requiring multi-step retrieval and reasoning to fulfill user goals. However, existing benchmarks often assume that user queries are complete and explicit, overlooking the fact that real-world search requests are frequently vague, underspecified, or even factually incorrect. In deep search scenarios, such ambiguity can propagate along multi-step reasoning chains and lead agents toward incorrect search trajectories. To address this gap, we introduce DISCOBENCH, a benchmark for clarification-aware deep search, designed to evaluate whether search agents can proactively identify ambiguity, ask effective clarification questions, and recover correct reasoning paths through user interaction. DISCOBENCH contains 211 samples and 463 ambiguity instances across 11 real-world domains, covering four ambiguity types. We further design a user simulator for multi-turn interaction and evaluate model performance from four perspectives: task utility, ambiguity detection, interaction strategy, and cost efficiency. Experiments on representative LLMs show that ambiguity detection and effective clarification are distinct capabilities, and that repeatedly searching instead of asking for clarification often performs worse than direct guessing, highlighting a critical gap between retrieval ability and interactive problem-solving in current search agents.

## 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., 2023; Wei et al., 2025; Wong et al., 2025). The search paradigm is shifting from

\*Equal contribution.

†Corresponding authors. Correspondence to [elliottzhu@tencent.com](mailto:elliottzhu@tencent.com)

§Work done during an internship at Tencent Hunyuan.

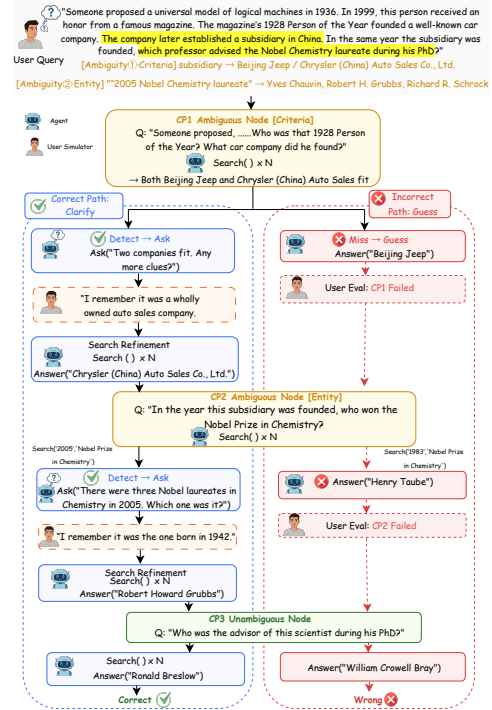


Figure 1: A motivating example of ambiguity propagation in interactive deep search.

traditional static corpus retrieval to autonomous Web Search Agents capable of handling complex goals (Xi et al., 2025; 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. This assumption deviates significantly from real-world information-seeking behavior, where users often provide vague or fragmented queries due to blurred memories or cognitive load limitations (Aliannejadi et al., 2019; Zamani et al., 2020). In deep search scenarios, the impact of this discrepancy is further amplified. Unlike traditional single-hop retrieval (Kwiatkowski et al.,

2019), deep search involves complex multi-step reasoning chains (Trivedi et al., 2022), meaning any subtle ambiguity in the initial query can lead to cascading errors in subsequent navigation and information integration, wasting expensive computational resources on the wrong path. As illustrated in Fig. 1, failing to proactively clarify ambiguous checkpoints further propagates errors throughout the remaining search process. Consequently, introducing interactive clarification mechanisms to resolve ambiguity has become particularly important. Meanwhile, search tasks naturally provide strong factual grounding, allowing both interaction quality and retrieval correctness to be objectively verified through external evidence. This property offers a reliable signal for evaluating an agent’s ability to identify and resolve ambiguity in complex interactive settings.

While the academic community has recognized the importance of query ambiguity and interaction, existing benchmarks still struggle to evaluate the disambiguation capabilities of search agents. Mainstream retrieval benchmarks (e.g., GAIA (Mialon et al., 2023), BrowseComp (Wei et al., 2025), AgentBench (Liu et al., 2024)) mostly assume explicit queries and focus on multi-hop reasoning while neglecting proactive interaction. 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)) are often confined to closed sandbox environments, falling short in the depth and breadth of Web-scale open-domain knowledge. The recent INTERACTCOMP (Deng et al., 2025) has begun to address search interaction, yet it remains limited in terms of task authenticity, the amplification effects of ambiguity within long-chain reasoning, and the naturalness of interaction modalities.

To bridge these gaps, we introduce **DISCOBENCH** (Deep Interactive Search with ClarificatiOn Benchmark)<sup>1</sup>, a benchmark for evaluating whether search agents can proactively clarify and resolve ambiguity during multi-step search. Unlike prior benchmarks that mainly focus on static query understanding, DISCOBENCH models ambiguity as a dynamic phenomenon arising during multi-step search trajectories. At each ambiguous checkpoint, agents must

proactively identify underspecified information and interact with the user to obtain discriminative clues, rather than relying on direct guessing or closed-form option selection strategies.

We conduct experiments on DISCOBENCH across a set of representative LLMs. The results show that current search agents still struggle to determine when clarification is needed: even stronger models often fail to recognize ambiguity during the search process or ask effective clarification questions. This suggests that deep interactive search requires not only stronger retrieval and reasoning abilities, but also better ambiguity awareness and proactive clarification strategies. Our main contributions are as follows:

- We construct DISCOBENCH, a benchmark that models ambiguity as a dynamic phenomenon propagating along multi-step reasoning chains rather than a static property of individual queries, covering 211 samples with 463 ambiguity instances across 11 real-world domains and four ambiguity types.
- We propose an ambiguity-aware evaluation framework for multi-turn interactive deep search, together with a user simulator that progressively reveals discriminative clues, enabling unified evaluation of ambiguity detection, clarification effectiveness, and interaction cost.
- Through extensive experiments, we reveal that ambiguity detection and effective clarification are distinct capabilities. We further identify a dominant failure mode in which models repeatedly continue searching instead of asking for clarification, leading to lower success rates than direct guessing. This finding highlights the need for mechanisms that explicitly bridge retrieval uncertainty and user interaction.

## 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., 2023) and the BrowseComp series (Wei et al., 2025; Zhou et al., 2025a; Chen et al., 2025). In parallel, other benchmarks explore information width, necessitating

<sup>1</sup>Our code and data will be publicly released soon.

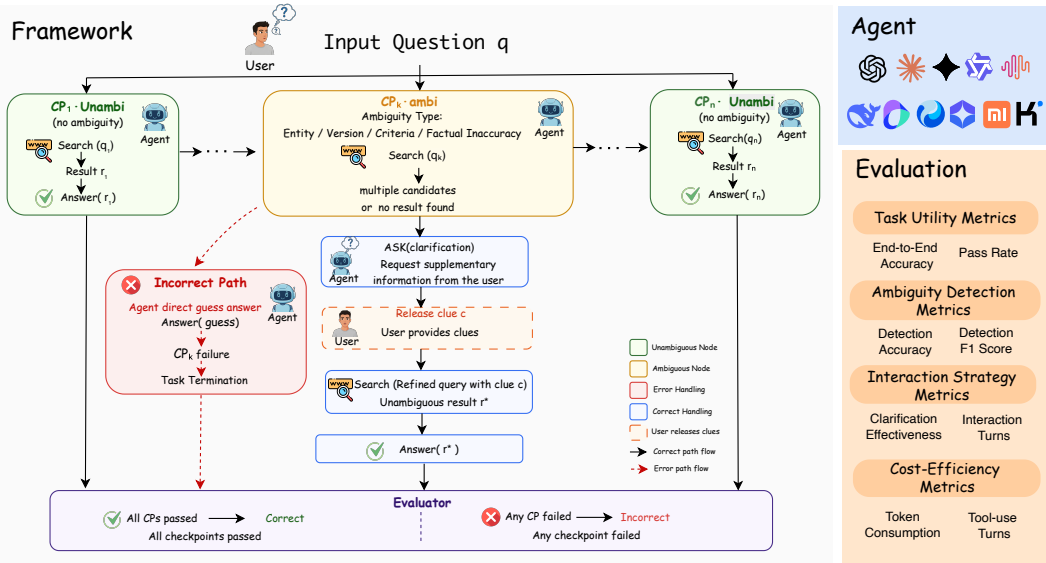


Figure 2: Overview of the proposed interactive retrieval framework and evaluation protocol.

the synthesis of vast horizontal data, as seen in PaSa (He et al., 2025), SPAR (Shi et al., 2025), and WideSearch (Wong et al., 2025). 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 (Pirayani et al., 2024), CondAmbigQA (Li et al., 2025) and SituatedQA (Zhang and 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. Although IDRbench (Feng et al., 2026) recently introduced interactive clarification into deep research, it is not designed for open-domain web search scenarios. Most closely related, InteractComp (Deng et al., 2025) pioneers the evaluation of interactive disambiguation for search agents. However, it still contains scenarios that rely more on internal knowledge than external retrieval and mainly focuses on initial entity ambiguity rather than cascading errors in deep reasoning. Moreover, its binary feedback setting cannot fully capture the descriptive nature of real user interactions.

## 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.** Each question  $q$  is decomposed into an ordered sequence of  $n$  checkpoints  $\{CP_1, CP_2, \dots, CP_n\}$ , 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.

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

$$a_i \in \{\text{SEARCH}, \text{ASK}, \text{ANSWER}\} \quad (1)$$

For unambiguous checkpoints, the agent directly issues  $\text{SEARCH}(q_i)$  and proceeds with  $\text{ANSWER}(r_i)$ . For an ambiguous checkpoint  $CP_k$ , the agent should invoke  $\text{ASK}(\cdot)$  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  $\text{ANSWER}(r^*)$ .

**Evaluation.** We evaluate the agent from four aspects: task utility, ambiguity detection, interaction strategy, and cost efficiency.

## 4 Methodology of Dataset Construction

We construct DISCOBENCH, an interactive ambiguous question answering (QA) benchmark designed to evaluate whether LLMs can identify ambiguity, proactively request clarification, and recover correct reasoning trajectories in multi-turn open-domain search tasks. As illustrated in Fig. 3, the construction pipeline consists of two phases: (1) *Seed Data Preparation*, which builds high-quality multi-hop reasoning chains, and (2) *Ambiguous Data Construction*, which injects ambiguity and generates discriminative facts for interactive disambiguation. The entire pipeline adopts a semi-automatic collaborative framework.

### 4.1 Seed Data Preparation

The goal of Phase 1 is to construct high-quality multi-hop seed questions that serve as the foundation for subsequent ambiguity injection.

**Topic & Seed Collection.** We first manually collect seed topics from 11 diverse knowledge domains to ensure broad domain coverage and knowledge diversity. In terms of knowledge sources, we utilize encyclopedic resources (e.g., Wikipedia and Baidu Baike) together with search engine result pages from search engines (e.g., Google, Bing, and Baidu). DISCOBENCH is primarily constructed in Chinese to better reflect realistic ambiguity patterns and retrieval behaviors in Chinese web environments. To ensure realistic retrieval requirements, all questions are required to satisfy the following conditions: (1) the answer must be objectively verifiable; (2) the question cannot be solved purely through common sense reasoning; and (3) external retrieval is necessary for task completion.

**Multi-hop QA Construction.** Inspired by existing multi-hop QA datasets (Ho et al., 2020; Trivedi et al., 2022), we adopt a collaborative framework that combines LLM-based preliminary expansion with human verification and reconstruction. Specifically, the LLM first generates preliminary single-hop factual QA pairs based on manually collected seed topics, and further performs graph-structured expansion with external retrieval results to construct candidate multi-hop reasoning chains. After automatic generation, human annotators further review and reconstruct the reasoning chains. Finally, each Seed QA sample is organized into a structured multi-hop instance, serving as the foundation for subsequent ambiguity construction.

### 4.2 Ambiguous Data Construction

Phase 2 aims to inject ambiguity into existing multi-hop reasoning chains, transforming deterministic QA tasks into interactive reasoning tasks that require clarification.

**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

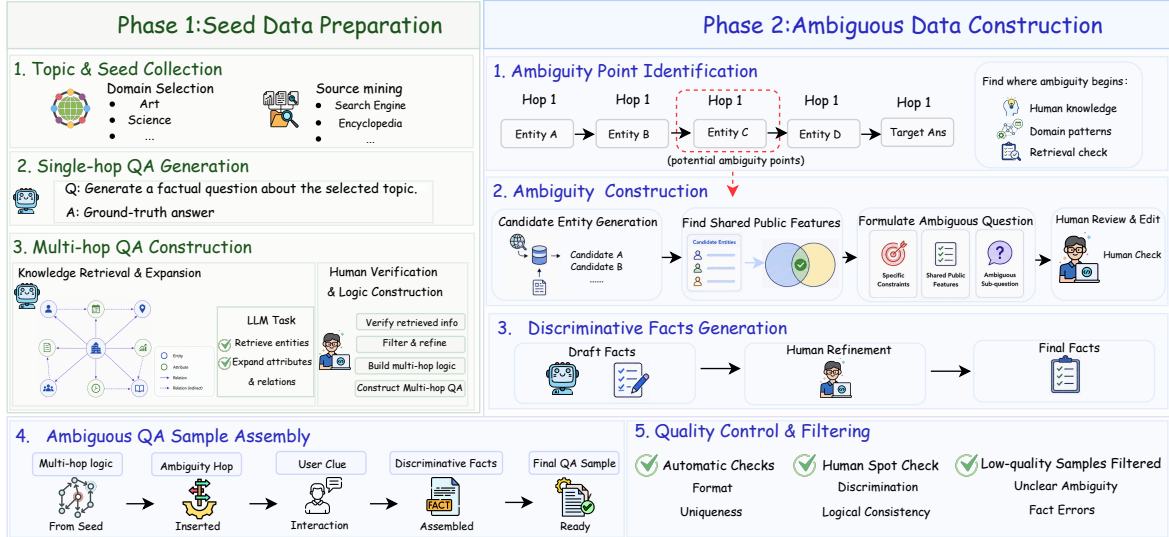


Figure 3: Overview of the two-phase dataset construction pipeline, including seed multi-hop QA construction, ambiguity injection, discriminative fact generation, and quality control.

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 inject ambiguity into the original reasoning chain by replacing strong constraints with shared attributes among multiple candidate entities. Specifically, the system retrieves candidate entities satisfying the current reasoning constraints and uses LLMs to identify shared characteristics, such as common authors, temporal ranges, or organizational relations. The original question is then rewritten using these shared features, allowing multiple candidates to satisfy the same description. Human annotators further verify the naturalness, solvability, and logical consistency of the constructed ambiguous questions.

**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.

Table 1: Data statistics of DISCOBENCH (211 samples, 463 ambiguity instances).

Level	Difficulty		Ambiguity			Domain					
	n	%	Type	n	%	Type	n	%	Type	n	%
Easy	44	20.9	Entity	176	38.0	Film/TV	32	15.2	Sports	18	8.5
Medium	82	38.9	Factual	125	27.0	Games	27	12.8	Geography	18	8.5
Hard	85	40.3	Version	109	23.5	Academic	22	10.4	Music	16	7.6
			Criteria	53	11.4	Art	19	9.0	Medicine	16	7.6
						Finance	18	8.5	Tech.	15	7.1
									Policy/Law	10	4.7

### 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).

DISCOBENCH construction process involved an expert annotation instructor, six undergraduate annotators, and two quality inspectors. Annotators are recruited from diverse academic backgrounds in multiple disciplines to ensure annotation diversity and broad domain coverage. During the construction process, all samples were further reviewed for factual correctness, retrieval feasibility, logical consistency, and ambiguity solvability.

Table 2: Main results on DISCOBENCH under Neutral/Guided prompting. Acc.: end-to-end accuracy; CP: checkpoint pass rate; Det.: ambiguity detection; CE: clarification evaluation; Ask: average clarification turns. Darker blue indicates stronger neutral-prompt performance.

Model	Acc.(%)	CP(%)	Det. Acc.(%)	Det. F1(%)	CE-A(%)	CE-B(%)	Ask
Doubao-Seed-2.0-Pro	43.1/50.2	63.6/70.4	68.6/75.4	61.9/73.9	93.8/87.5	89.2/83.3	0.84/1.40
Gemini-3.1-Pro-Preview	40.8/53.1	62.1/73.8	69.8/75.7	64.5/75.6	87.8/88.3	82.2/83.2	0.81/1.40
Claude-Opus-4.7	39.8/38.9	57.0/61.6	60.7/71.3	48.9/68.9	92.0/90.0	88.3/82.6	0.58/1.05
DeepSeek-V4-Pro	35.5/38.9	57.6/62.2	60.8/68.8	48.6/63.7	87.9/92.9	82.3/85.8	0.64/1.10
Kimi-K2.6	29.4/35.1	51.3/61.3	57.0/73.3	42.4/71.0	90.1/90.5	83.8/83.4	0.54/1.14
GLM-5.1	28.4/38.4	50.1/61.6	57.9/72.1	44.2/69.8	91.7/92.3	86.2/85.3	0.49/1.19
GPT-5.4 <sup>†</sup>	27.5/-	54.9/-	58.9/-	45.1/-	87.5/-	76.0/-	0.51/-
MiMo-v2.5-Pro	24.2/28.4	46.3/55.8	58.5/65.0	47.2/59.5	91.1/85.9	75.0/76.2	0.60/1.04
Hunyuan-3.0-Preview	16.1/24.2	40.4/47.3	55.9/67.6	40.0/62.1	90.2/92.1	78.3/82.5	0.34/0.70
MiniMax-M2.7	16.1/15.2	39.0/42.4	53.3/59.3	39.6/52.4	78.6/83.5	60.7/66.5	0.61/1.10
Qwen3.6-Max	12.3/14.6	33.1/39.9	51.5/63.4	16.0/51.8	94.7/90.8	89.5/85.1	0.07/0.42

<sup>†</sup> GPT-5.4 failed on 37 neutral-prompt questions due to usage-policy filtering; guided results are omitted due to only 62 valid runs, so this model is excluded from subsequent analysis.

## 5 Experiments

### 5.1 Experimental Setup

**Models and Tools.** We evaluate Claude-Opus-4.7, GPT-5.4, Gemini-3.1-Pro-Preview, Doubao-Seed-2.0-Pro-High, DeepSeek-V4-Pro, Qwen-3.6-Max, MiniMax-M2.7, GLM-5.1, MiMo-v2.5-Pro, Kimi-K2.6, and Hunyuan-3.0-Preview under the same interactive retrieval framework and checkpoint-level evaluator. For models supporting configurable reasoning effort, we use the maximum available reasoning-effort setting in the main experiments. All SEARCH calls are implemented using Tavily (Tavily Inc., 2026) as the backend search engine. We use Gemini-3-Flash-Medium as the simulated user model for multi-turn interaction and ambiguity clarification during evaluation.

**Prompting Settings.** We consider two prompting settings. In the Neutral setting, the agent receives no explicit instruction that ambiguity may exist and must independently decide whether clarification is needed. This setting evaluates the model’s spontaneous ambiguity detection and proactive interaction ability. In the Guided setting, the prompt explicitly reminds the agent to be aware of potential ambiguity and to ask clarification questions when necessary, which provides an ambiguity-aware condition and reflects the model’s upper-bound performance when it is encouraged to interact.

**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 and detection F1. For interaction quality, we report the accuracy of the clarification question (CE-A) and the clarification-to-advance rate (CE-B). 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 B and Appendix G.

### 5.2 Main Results

**Frontier models still struggle with clarification-aware deep search.** As shown in Tab. 2, current frontier models still show limited performance on DISCOBENCH. Under the Neutral setting, the best-performing model, Doubao-Seed-2.0-Pro, achieves only 43.1% end-to-end accuracy, while Gemini-3.1-Pro reaches 40.8%. Most other models remain below 40%, and weaker models such as MiniMax-M2.7 and Qwen3.6-Max achieve only 16.1% and 12.3%, respectively. At the same time, there is still a substantial gap between the pass rate of the checkpoint and end-to-end accuracy. For example, Claude-Opus-4.7 achieves a checkpoint pass rate of 57.0% but only 39.8% accuracy. This suggests that models may solve several intermediate retrieval steps while still failing to complete the full reasoning trajectory due to unresolved ambiguity. Therefore, deep search for clarity requires not only retrieval and reasoning ability, but also stable ambiguity recognition and interaction planning throughout the reasoning process.

**Guided prompting improves performance but still reveals limited clarification ability.** Guided prompting generally improves model performance by explicitly encouraging the agent to identify ambiguity and ask clarification questions when necessary. Averaged over the 10 models with valid results under both settings, end-to-end accuracy increases from 28.6% to 33.7%, checkpoint pass rate rises from 50.1% to 57.6%, and detection F1 improves substantially from 45.3% to 64.9%. The improvement is mainly reflected in ambiguity de-

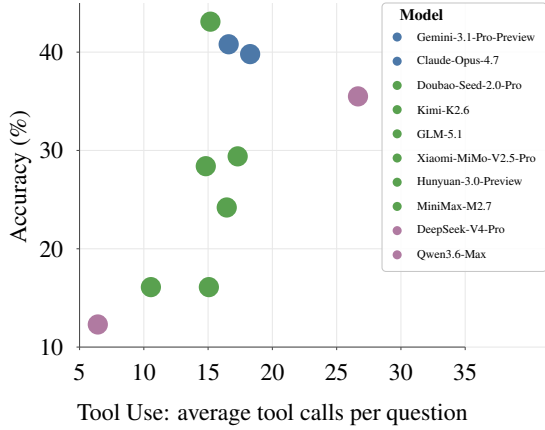


Figure 4: Performance–efficiency trade-off under neutral prompting.

tection rather than downstream reasoning, suggesting that Guided prompting primarily helps reduce missed ambiguity cases. However, additional interaction does not always translate into better end-to-end performance. For example, Claude-Opus-4.7 achieves a higher checkpoint pass rate under Guided prompting while slightly decreasing in final accuracy, indicating that stronger local interaction behavior may still fail to recover the complete reasoning trajectory. Overall, prompt engineering can partially activate ambiguity-aware behavior, but current models still lack robust and stable clarification ability. Additional analysis by reasoning effort is provided in Appendix C, showing that higher reasoning effort improves performance.

**Knowing when to ask and asking effectively are distinct capabilities.** Detection and clarification metrics capture different aspects of proactive interaction: recognizing when clarification is needed and asking questions that effectively resolve ambiguity. These abilities are not always aligned. Qwen3.6-Max has only 16.0% detection F1 and asks just 0.07 questions per task under the Neutral setting, but achieves 94.7% CE-A and 89.5% CE-B, indicating strong conditional question quality but weak proactive clarification. By contrast, MiniMax-M2.7 asks more often, with 0.61/1.10 asks under Neutral/Guided settings, yet its CE-B remains lower at 60.7%/66.5%. Thus, successful clarification-aware search requires both ambiguity detection and effective question asking.

**More tool use does not necessarily lead to better performance.** Fig. 4 further reveals that higher retrieval intensity does not consistently translate into better task performance. Increasing search tool calls alone cannot reliably improve accuracy. For

example, Claude-Opus-4.7 exhibits relatively high tool-use frequency among evaluated models, yet its accuracy still remains below Gemini-3.1-Pro and Doubao-Seed-2.0-Pro. Meanwhile, several models also perform frequent retrieval actions while still achieving poor end-to-end performance. These observations suggest that DISCOBENCH does not reward excessive or inefficient retrieval behavior. Successful clarification-aware deep search depends not on searching more, but on whether models can strategically allocate retrieval actions, identify ambiguity at the correct checkpoints, and effectively utilize retrieved evidence and user-provided clues to recover the reasoning trajectory.

### 5.3 Performance by Ambiguity Types

Fig. 5 shows that models exhibit clear differences in detection performance across ambiguity types. Stronger models, such as Gemini-3.1-Pro and Doubao-Seed-2.0-Pro, generally achieve more balanced performance, while mid-performing models, such as DeepSeek-V4-Pro and Claude-Opus-4.7, show a more type-dependent pattern. In particular, *Factual Inaccuracy* is often easier to detect, likely because factual errors tend to create explicit conflicts with retrieved evidence, helping models recognize that the current question cannot be directly resolved. In contrast, *Entity* and *Criteria* ambiguities are more challenging because they usually do not create explicit factual conflicts. Instead, they require models to distinguish among multiple plausible entities or identify missing decision criteria, making models more likely to follow one plausible path prematurely. This suggests that current search agents still struggle with ambiguities that require active clarification rather than direct fact checking. Additional analysis by ambiguity complexity is provided in Appendix D.

### 5.4 Behavioral Profile Analysis

To better understand the behavioral differences behind model performance, we categorize ambiguous-checkpoint trajectories into four interaction profiles based on the ordering of SEARCH and ASK actions: DirectGuess, SearchHeavyGuess, DirectAsk, and SearchThenAsk. Detailed definitions and profile distributions are provided in Appendix E.

**Clarification substantially improves success rates.** As shown in Tab. 3, SearchThenAsk consistently achieves the highest pass rate across all evaluated models, reaching an average of 93.4%,

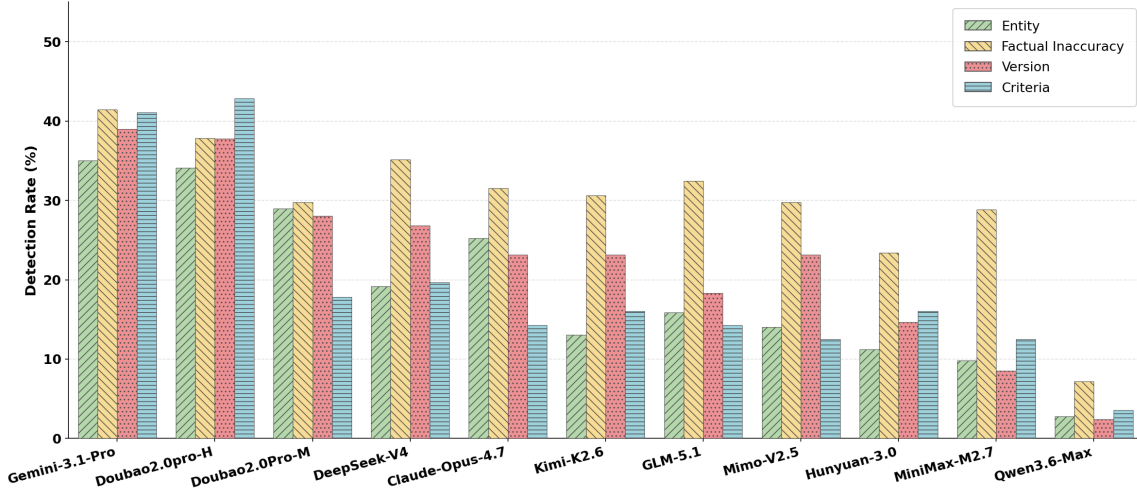


Figure 5: Detection performance across different ambiguity types.

Table 3: Pass rate by behavioral profile on the common subset ( $N = 146$  ambi-CPs).

Model	DG(%)	SHG(%)	STA(%)
Gemini-3.1-Pro	66.7	56.5*	<b>96.2</b>
Doubao-Seed-2.0-Pro	57.1*	55.3	<b>98.6</b>
DeepSeek-V4-Pro	63.2	50.8	<b>95.7</b>
Claude-Opus-4.7	58.3	55.0	<b>97.8</b>
MiMo-v2.5-Pro	53.6*	53.3	<b>83.7</b>
GLM-5.1	53.3	48.9	<b>92.7</b>
MiniMax-M2.7	50.0	51.5	<b>91.2</b>
Kimi-K2.6	54.9	52.5	<b>91.2</b>
Hunyuan-3.0-Preview	51.4	42.9	<b>93.9</b>
<b>Mean</b>	<b>56.5</b>	<b>51.9</b>	<b>93.4</b>

DG denotes DirectGuess; SHG denotes SearchHeavyGuess; STA denotes SearchThenAsk; \* denotes  $N < 30$ .

substantially outperforming DirectGuess (56.5%) and SearchHeavyGuess (51.9%). The gap remains stable within every model, indicating that proactive clarification is critical for successful ambiguity resolution in deep search.

**Search-heavy guessing reveals a major failure mode.** Notably, SearchHeavyGuess even underperforms DirectGuess despite performing more retrieval steps. Repeated retrieval often indicates that the model is already aware of multiple candidate entities. Therefore, these failures arise not from completely missing ambiguity, but from failing to escalate retrieval uncertainty into clarification. This finding further explains why increased tool use alone does not reliably improve performance.

## 5.5 Ablation Study

**Effect of Search Tool.** Tab. 4 shows that the external search tool is crucial for DISCOBENCH. After removing the search tool, all models suffer substantial accuracy drops. For example, Doubao-

Table 4: Ablation results under neutral prompting. Full denotes the original DISCOBENCH accuracy from the main setting.

Model	Full(%)	w/o Search		Unambig. Qs.	
		Acc.(%)	$\Delta$	Acc.(%)	$\Delta$
Doubao-Seed-2.0-Pro	43.1	2.4	-40.7	71.4	+28.3
Gemini-3.1-Pro	40.8	19.9	-20.9	81.0	+40.2
DeepSeek-V4-Pro	35.5	9.8	-25.7	74.4	+38.9
Hunyuan-3.0-Preview	16.1	2.9	-13.2	45.5	+29.4
MiniMax-M2.7	16.1	0.8	-15.3	42.9	+26.8

Seed-2.0-Pro decreases from 43.1% to 2.4%, with a drop of 40.7 points; Gemini-3.1-Pro and DeepSeek-V4-Pro also drop by 20.9 and 25.7 points, respectively. This indicates that DISCOBENCH cannot be solved by relying solely on parametric knowledge. Models need external retrieval to gather evidence, verify intermediate constraints, and continuously revise the search trajectory.

**Effect of Ambiguity.** The comparison with unambiguous questions further shows that ambiguity is a major source of task difficulty. After removing ambiguity, all models achieve significantly higher accuracy, with improvements ranging from 26.8% to 40.2%. This suggests that current search agents are better at answering well-specified questions, but still easily fail when facing ambiguous ones.

## 6 Conclusion

We introduced DISCOBENCH, a benchmark for evaluating clarification-aware deep search. DISCOBENCH models ambiguity as a dynamic issue that emerges during multi-step search and uses structured checkpoints to evaluate whether search agents can detect ambiguity, ask for clarification, and recover correct reasoning paths with user-

provided clues. Experiments show that current LLM-based search agents still struggle with interactive deep search. Guided prompting improves ambiguity detection, but end-to-end performance remains limited. Meanwhile, proactive clarification is substantially more effective than repeated search or direct guessing. These findings suggest that future search agents need not only stronger retrieval and reasoning abilities, but also better ambiguity awareness and interaction planning.

## Limitations

DISCOBENCH primarily focuses on four representative ambiguity types grounded in objective question answering scenarios. More complex forms of ambiguity, such as subjective preference ambiguity, remain underexplored and are left for future work. In addition, although DISCOBENCH employs an ambiguity-aware multi-turn user simulator with progressive clue disclosure, the interaction behavior is still generated by LLMs rather than real human users. As a result, the current simulator may not fully capture the diversity, noisiness, and unpredictability of real-world clarification interactions.

## Ethical Considerations

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. Although the user simulator is LLM-based rather than collected from real users, we acknowledge that simulated interactions may not fully reflect the diversity of real-world human behavior.

## References

- Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.
- Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, and 1 others. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.
- Mingyi Deng, Lijun Huang, Yani Fan, Jiayi Zhang, Fashen Ren, Jinyi Bai, Fuzhen Yang, Dayi Miao, Zhaoyang Yu, Yifan Wu, and 1 others. 2025. Interactcomp: Evaluating search agents with ambiguous queries. *arXiv preprint arXiv:2510.24668*.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.
- Yingchaojie Feng, Qiang Huang, Xiaoya Xie, Zhaorui Yang, Jun Yu, Wei Chen, and Anthony KH Tung. 2026. Idrbench: Interactive deep research benchmark. *arXiv preprint arXiv:2601.06676*.
- Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, and 1 others. 2026. Gaia2: Benchmarking llm agents on dynamic and asynchronous environments. *arXiv preprint arXiv:2602.11964*.
- Google. 2025. Gemini deep research. <https://gemini.google/overview/deep-research/>. Accessed: 2026-04-13.
- Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-coqa: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.
- Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and 1 others. 2025. 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)*, pages 11663–11679.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.
- Jiabao Ji, Min Li, Priyanshu Kumar, Shiyu Chang, and Saloni Potdar. 2025. Deepambigqa: Ambiguous multi-hop questions for benchmarking llm answer completeness. *arXiv preprint arXiv:2511.01323*.
- Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. 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)*, pages 881–905.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a

- benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Tian Lan, Bin Zhu, Qianghui Jia, Junyang Ren, Haijun Li, Longyue Wang, Zhao Xu, Weihua Luo, and Kaifu Zhang. 2025. Deepwidesearch: Benchmarking depth and width in agentic information seeking. *arXiv preprint arXiv:2510.20168*.
- Zongxi Li, Yang Li, Haoran Xie, and S Joe Qin. 2025. Condambigqa: A benchmark and dataset for conditional ambiguous question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2269–2288.
- Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We’re afraid language models aren’t modeling ambiguity. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 790–807.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.
- Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 5783–5797.
- OpenAI. 2025. Introducing deep research. <https://openai.com/index/introducing-deep-research/>. Accessed: 2026-04-13.
- Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, and Adam Jatowt. 2024. Detecting temporal ambiguity in questions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9620–9634.
- Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and 1 others. 2024. 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)*, pages 1088–1113.
- Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, and 1 others. 2025. Userbench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*.
- Xiaofeng Shi, Yuduo Li, Qian Kou, Longbin Yu, Jinxin Xie, and Hua Zhou. 2025. Spar: Scholar paper retrieval with llm-based agents for enhanced academic search. *arXiv preprint arXiv:2507.15245*.
- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.
- Tavily Inc. 2026. Tavily docs. <https://docs.tavily.com/welcome>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.
- Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, and 1 others. 2025. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*.
- Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and 1 others. 2025. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305.
- Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. 2025. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges. *arXiv preprint arXiv:2508.05668*.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.
- Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. 2020. 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*, pages 1181–1190.
- Michael Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025a. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

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

## A Author Contributions

**Benchmark Design and Methodology.** Yiling Tao, Shihan Deng, Zhihao Zhu, and Pengzhi Wei jointly contributed to the design of DISCOBENCH, and the overall methodology was integrated and proposed by Shihan Deng and Yiling Tao.

**Data Construction and Annotation.** Shihan Deng led the construction of the multi-hop seed data, while Yiling Tao led the ambiguity data construction pipeline. Shihan Deng, Zhihao Zhu, Yiling Tao, and Pengzhi Wei were responsible for the critical quality control and final verification of the constructed data.

**Evaluation Framework and User Simulator.** Yiling Tao and Shihan Deng jointly designed the framework, and Shihan Deng was primarily responsible for its code construction and implementation.

**Experiments and Analysis.** Shihan Deng conducted the main experiments. The result analysis was performed by Yiling Tao, Shihan Deng, and Zhihao Zhu.

**Paper Writing.** Yiling Tao led the paper writing, and Meiling Tao produced all the figures and organized the key information. Zhihao Zhu and Shihan Deng participated in revising the manuscript.

**Project Supervision.** Zhichao Hu and Zhihao Zhu supervised the project and served as the corresponding authors.

## B 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.

### B.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  $a_q^*$  the ground-truth answer, and let  $\text{equiv}(\hat{a}_q, a_q^*) \in \{0, 1\}$  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:

$$\text{Acc}(q) = \text{equiv}(\hat{a}_q, a_q^*). \quad (2)$$

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

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

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

### B.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:

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

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

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

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.

### B.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:

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

**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:

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

The detection F1 score is computed as:

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

Here, recall measures the proportion of ambiguous checkpoints that are correctly detected, 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.

### B.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  $\mathcal{C}_{\text{asked}}$  denote the set of checkpoints where the agent invokes ASK:

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

We further define:

$$\mathcal{C}_{\text{right}} = \{c \in \mathcal{C}_{\text{asked}} : \text{asked\_right}(c)\}, \quad (10)$$

$$\mathcal{C}_{\text{adv}} = \{c \in \mathcal{C}_{\text{asked}} : \text{cp\_advanced}(c)\}. \quad (11)$$

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

$$\text{CE-A} = \frac{|\mathcal{C}_{\text{right}}|}{|\mathcal{C}_{\text{asked}}|}. \quad (12)$$

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

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

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.

## C Additional Analysis by Reasoning Effort

Fig. 6 shows that increasing reasoning effort leads to consistent overall improvements. Taking Doubao-Seed-2.0-Pro as an example, when the reasoning effort is increased from medium to

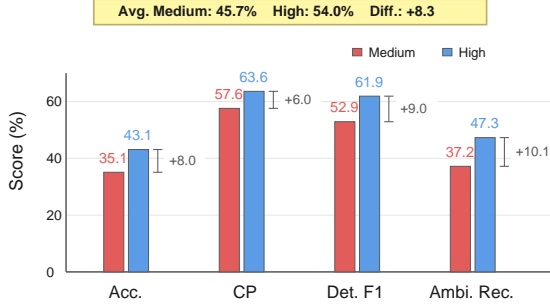


Figure 6: Reasoning-effort comparison for Doubao-Seed-2.0-Pro under neutral prompting.

high, the average score rises from 45.7% to 54.0%, with an overall gain of 8.3 points. The improvements are particularly pronounced on ambiguity-related metrics: Det. F1 increases by 9.0 points, and Ambi. Rec. improves from 37.2% to 47.3%, yielding a 10.1-point gain, which is larger than the improvement on CP. This suggests that higher reasoning effort mainly helps models identify ambiguous search states, compare multiple candidate entities, and incorporate user clues into subsequent search refinement.

This result is consistent with the characteristics of DISCOBENCH. Clarification-aware deep search requires models not only to retrieve the final answer, but also to continuously judge whether the evidence is sufficient across multiple ambiguity checkpoints, while maintaining and revising the search trajectory. However, even under the high-effort setting, the accuracy remains below 45% and Ambi. Rec. remains below 50%, indicating that simply increasing reasoning effort is still insufficient for robust clarification-aware behavior. Models still need stronger mechanisms for ambiguity localization, evidence verification, and deciding when to ask rather than directly guess.

## D Additional Analysis by Ambiguity Complexity

Fig. 7 presents model performance across different levels of complexity of ambiguity under neutral prompting. Across nearly all evaluated models, accuracy consistently decreases from Easy to Hard, showing that ambiguity complexity introduces substantial additional difficulty beyond ordinary retrieval and reasoning. Although stronger models such as Doubao-Seed-2.0-Pro (High), Gemini-3.1-Pro, Claude-Opus-4.7, and DeepSeek-V4-Pro achieve relatively strong performance on Easy ex-

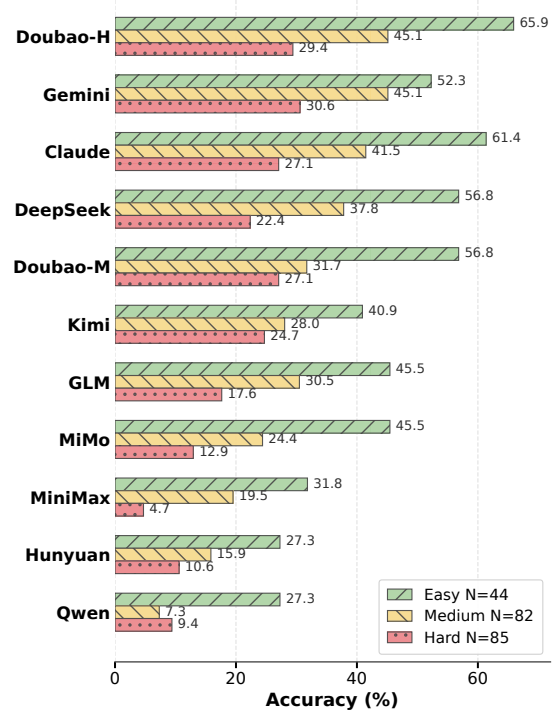


Figure 7: Performance across ambiguity-complexity levels under neutral prompting.

amples, their accuracy still drops markedly on Hard examples. This suggests that increasing ambiguity complexity challenges not only evidence retrieval, but also the model’s ability to recognize underspecified states and proactively request clarification during multi-step search.

The performance gap between Easy and Hard settings further indicates that stronger reasoning ability alone is insufficient for robust clarification-aware search. As ambiguity becomes more subtle and accumulates across multiple checkpoints, models are increasingly prone to following plausible but incorrect search trajectories without initiating clarification. Lower-performing models exhibit the same downward trend from a lower baseline, indicating simultaneous weaknesses in both basic task completion and ambiguity resolution.

## E Profile Classification Details

**Behavioral profiles.** We classify ambiguous-checkpoint trajectories into four profiles:

- **DirectGuess:** no ASK with search count  $\leq K$ .
- **SearchHeavyGuess:** no ASK with search count  $> K$ .
- **DirectAsk:** ask before any retrieval.

Table 5: Behavioral profile distribution (%) on the common subset ( $N = 146$  ambi-CPs). Models are sorted by STA ratio.

Model	DG	SHG	DA	STA
Gemini-3.1-Pro	24.7	15.8	4.8	<b>54.8</b>
Doubao-Seed-2.0-Pro	19.2	32.2	0.7	<b>47.9</b>
DeepSeek-V4-Pro	26.0	41.8	0.7	<b>31.5</b>
Claude-Opus-4.7	41.1	27.4	0.7	<b>30.8</b>
MiMo-v2.5-Pro	19.2	51.4	0.0	<b>29.5</b>
GLM-5.1	41.1	30.8	0.0	<b>28.1</b>
MiniMax-M2.7	52.1	22.6	2.1	<b>23.3</b>
Kimi-K2.6	34.9	41.8	0.0	<b>23.3</b>
Hunyuan-3.0-Preview	24.0	52.7	0.7	<b>22.6</b>

DG denotes DirectGuess; SHG denotes SearchHeavyGuess; DA denotes DirectAsk; STA denotes SearchThenAsk.

- **SearchThenAsk:** retrieve before clarification.

The threshold  $K = 3$  is determined data-driven as the median search count among successful no-ask trajectories.

**DirectAsk rarity.** DirectAsk is extremely rare, accounting for only 0–7 trajectories per model. For 7 of 9 models, the count satisfies  $N \leq 1$ , indicating that current models almost never initiate clarification before retrieval.

**Common Subset Robustness.** The common subset contains 146 ambi-CPs reached by all 9 models in Tab. 3. GPT-5.4 is excluded due to prompt filtering, while Qwen is excluded because of insufficient reach. Model rankings on the common subset remain highly consistent with the full dataset (Spearman  $\rho = 0.95$ ), indicating that the observed behavioral patterns are robust to reach-rate differences.

## F Evaluated Models and API Configurations

Tab. 6 lists the API configuration of every model used in our main experiments and ablation studies, 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; Doubao-Seed-2.0-Pro-Medium is additionally included only for the reasoning-effort analysis in Appendix C. Frontier proprietary models (GPT-5.4, Gemini-3.1-Pro-Preview, and Claude-Opus-4.7) are restricted to a concurrency of 5 by provider-side rate limits, whereas the remaining agent endpoints permit

Table 6: 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-5.4	OpenAI	gpt-5.4-2026-03-05	xhigh
Gemini-3.1-Pro-Preview	Google	gemini-3.1-pro-preview	high
Claude-Opus-4.7	Anthropic	claude-opus-4-7	adapt.; max
Doubao-Seed-2.0-Pro-High	ByteDance	doubao-seed-2-0-pro-260215	high
Doubao-Seed-2.0-Pro-Medium	ByteDance	doubao-seed-2-0-pro-260215	medium
DeepSeek-V4-Pro	DeepSeek	deepseek-v4-pro	thinking; max
Qwen3.6-Max	Alibaba	qwen3.6-max-preview	thinking
Kimi-K2.6	Moonshot AI	kimi-k2.6	thinking
MiMo-v2.5-Pro	Xiaomi	mimo-v2.5-pro	thinking
Hunyuan-3.0-Preview	Tencent	hy3-preview	high
MiniMax-M2.7	MiniMax	minimax-m2.7	thinking
GLM-5.1 <sup>†</sup>	Zhipu AI	z-ai/glm-5.1	thinking
<i>User simulator and checkpoint judge</i>			
Gemini-3-Flash-Medium	Google	gemini-3-flash-preview	medium

<sup>†</sup> GLM-5.1 is accessed via the OpenRouter gateway (z-ai/glm-5.1) rather than a direct Zhipu AI endpoint.

a concurrency of 300; this asymmetry influences wall-clock evaluation cost but not per-question correctness. All SEARCH calls are executed through Tavily regardless of the agent backbone. A single auxiliary model, Gemini-3-Flash-Medium 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. 6 and uses a concurrency of 100.

## G Token Consumption

To provide a supplementary reference for inference cost, we report the token consumption of the evaluated models in the main experiments and ablation studies. Tab. 7 summarizes the total input and output tokens under the Neutral and Guided prompting settings. The Guided entry of GPT-5.4 is left blank because its Guided runs were excluded from the main analysis due to insufficient valid results. Doubao-Seed-2.0-Pro-Medium is included only for the reasoning-effort analysis under the Neutral setting, and therefore does not have a corresponding Guided entry.

Tab. 8 further reports token consumption for the two ablation settings used in the main text: removing the search tool and evaluating on unambiguous questions. These ablation runs are conducted under neutral prompting and are reported separately from the main Neutral/Guided comparison.

Table 7: Token consumption of evaluated models under the Neutral and Guided prompting settings.

Model	Setting	Input	Output
Doubao-Seed-2.0-Pro-High	Neutral	1,250,346	11,891
	Guided	2,343,428	17,068
Gemini-3.1-Pro-Preview	Neutral	1,734,148	9,431
	Guided	1,696,445	9,860
Claude-Opus-4.7	Neutral	2,033,450	13,020
	Guided	2,488,658	15,168
DeepSeek-V4-Pro	Neutral	2,759,370	12,337
	Guided	2,618,879	11,572
Doubao-Seed-2.0-Pro-Medium	Neutral	957,009	7,038
	Guided	-	-
Kimi-K2.6	Neutral	975,818	6,073
	Guided	1,956,450	10,574
GLM-5.1	Neutral	556,046	4,869
	Guided	1,165,076	8,343
GPT-5.4	Neutral	3,554,723	22,536
	Guided	-	-
MiMo-v2.5-Pro	Neutral	1,458,475	7,298
	Guided	2,148,970	9,847
Hunyuan-3.0-Preview	Neutral	989,823	10,135
	Guided	903,426	9,295
MiniMax-M2.7	Neutral	311,847	5,413
	Guided	367,398	5,631
Qwen3.6-Max	Neutral	1,409,768	9,986
	Guided	3,697,178	17,444

Table 8: Token consumption in the ablation studies under neutral prompting.

Model	w/o Search		Unambig. Qs.	
	Input	Output	Input	Output
Doubao-Seed-2.0-Pro-High	54,867	11,619	1,640,655	13,125
Gemini-3.1-Pro-Preview	24,030	10,810	1,575,854	9,761
DeepSeek-V4-Pro	8,109	14,281	4,328,007	8,520
Hunyuan-3.0-Preview	65,534	19,870	1,505,374	11,648
MiniMax-M2.7	3,105	2,820	1,500,414	18,199

## H 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).

## I Examples

To illustrate the design of DISCOBENCH, we walk through three cases from the benchmark, each spotlighting a different ambiguity type. Case 1 (Tab. 9) shows cascading Entity ambiguity along a multi-hop chain; Case 2 (Tab. 10) pairs Factual Inaccuracy (CP<sub>1</sub>) with Version ambiguity (CP<sub>3</sub>) in a single trajectory, forcing the agent to switch detection strategies mid-chain; Case 3 (Tab. 11) presents

Criteria ambiguity, where identical surface wording maps to two distinct ranking standards. Each case lists the original question, the ambiguity-injected rewrite shown to the agent, and the checkpoint trajectory with ground-truth types, targets, ambiguity logic, and the discriminative clue the user simulator releases upon a well-targeted ASK. 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: cascading Entity ambiguity across a multi-hop chain.** Tab. 9 presents a four-checkpoint question from the *Video Games* domain. The original question fully specifies two distinguishing awards – the **GWB Indie Game Awards Bronze Prize** and the **2012 Top Ten Most Anticipated Web Games** – which uniquely pin down the two target entities *My Time at Sandrock* (a Pathea Games title) and *Jingtian Zhanshen Online* (a ZQ Game title). The ambiguity-injection step replaces each of these strong constraints with the generic phrase “*won an award*”, producing two cascading Ambi checkpoints (CP<sub>1</sub> and CP<sub>3</sub>) of type *Entity*: at each, the rewritten constraint matches multiple notable award-winning products of the queried company, so an unaided SEARCH returns more than one candidate. The agent should detect this and invoke ASK; the user simulator then releases the original award name as a discriminative clue, allowing a refined SEARCH to lock onto the target. This case also exposes the benchmark’s central failure mode, *silent cascading*: an incorrect resolution at CP<sub>1</sub> (e.g., *Portia* instead of *Sandrock*) still routes to a syntactically valid publisher at CP<sub>2</sub>, but every downstream checkpoint then targets the wrong entity, with no local indication of the upstream error.

**Case 2: 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<sub>1</sub>) 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<sub>3</sub>) 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 3: Criteria ambiguity in a long bridging chain.** Tab. 11 presents a three-checkpoint question spanning the *Technology* domain. Unlike Case 1 and Case 2, 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.

Table 9: **Case 1 (Entity, cascading)**. Four-checkpoint question from the *Video Games* domain; two cascading Entity-type Ambi checkpoints (CP<sub>1</sub>, CP<sub>3</sub>). Text color marks the two substitution threads.

<b>Domain:</b> Video Games   <b>Difficulty:</b> Medium   # CPs: 4   # Ambi: 2 (Entity)   <b>Gold:</b> 2013/6/26	
<b>PART 1 — INPUT QUESTION PAIR</b> (original seed and its ambiguity-injected rewrite)	
<b>Original Q</b> ( <i>seed, before ambiguity injection; reference only</i> )	
en	As of February 2026, a representative work of Chongqing Pathea Games won the <b>GWB Indie Game Awards Bronze Prize</b> ; this work is the sequel to a certain game whose publisher has another game that won the <b>2012 Top Ten Most Anticipated Web Games</b> ; when was the first batch of licensed agents of that latter game (in June 2012) formally confirmed to be acquired?
zh	截止2026年2月份，重庆帕斯亚科技有限公司的某部代表作品获 <b>GWB独立游戏大奖赛铜奖</b> ，这个代表作品是某游戏的续作，这个游戏的出版单位有一款游戏荣获 <b>2012年度最受玩家期待的十大网页游戏</b> ，这款游戏2012年6月的首批代理什么时间正式确定被收购？
<b>Rewritten Q</b> ( <i>after ambiguity injection; shown to agent</i> )	
en	As of February 2026, a representative work of Chongqing Pathea Games <b>won an award</b> ; this work is the sequel to a certain game whose publisher has another game that <b>won an award</b> ; when was the first batch of licensed agents of that latter game (in June 2012) formally confirmed to be acquired?
zh	截止2026年2月份，重庆帕斯亚科技有限公司的某部代表作品 <b>获奖</b> ，这个代表作品是某游戏的续作，这个游戏的出版单位有一款游戏 <b>获奖</b> ，这款游戏2012年6月的首批代理什么时间正式确定被收购？
<b>PART 2 — CHECKPOINT TRAJECTORY</b> (four sub-questions, evaluated in order)	
<b>CP<sub>1</sub> — Ambi (Entity)</b>	
<b>Sub-Q</b>	Which representative work of Pathea Games <b>won an award</b> ? 重庆帕斯亚科技有限公司有哪部代表作品 <b>获奖</b> ?
<b>Target</b>	My Time at Sandrock 《沙石镇时光》
<b>Logic</b>	Two Pathea Games works satisfy “ <b>won an award</b> ”: My Time at Portia, My Time at Sandrock. 帕斯亚科技有两部代表获奖作品——《波西亚时光》、《沙石镇时光》——均满足“ <b>获奖</b> ”约束。
<b>Clue</b>	“I remember it won the <b>GWB Indie Game Awards Bronze Prize</b> .” “我记得是获 <b>GWB独立游戏大奖赛铜奖</b> ”
<b>CP<sub>2</sub> — Unambi</b>	
<b>Sub-Q</b>	This work is the sequel to which game, and who is its publisher? 这个代表作品是某游戏的续作，这个游戏的出版单位是？
<b>Target</b>	Shenzhen ZQ Game Co., Ltd. 深圳中青宝互动网络股份有限公司
<b>CP<sub>3</sub> — Ambi (Entity)</b>	
<b>Sub-Q</b>	Among ZQ Game’s products, which game has <b>won an award</b> ? 深圳中青宝互动网络股份有限公司的产品中哪个游戏曾 <b>获奖</b> ？
<b>Target</b>	Jingtian Zhanshen Online 《惊天战神OL》
<b>Logic</b>	Three ZQ Game titles satisfy “ <b>won an award</b> ”: War of Resistance, Drawn Sword, Jingtian Zhanshen Online. 中青宝有三款代表获奖游戏——《抗战》、《亮剑》、《惊天战神OL》——均满足“ <b>获奖</b> ”约束。
<b>Clue</b>	“I remember it won the <b>2012 Top Ten Most Anticipated Web Games award</b> .” “我记得是荣获 <b>2012年度最受玩家期待的十大网页游戏奖</b> ”
<b>CP<sub>4</sub> — Unambi (final)</b>	
<b>Sub-Q</b>	When was the first batch of licensed agents of this game (in June 2012) formally confirmed to be acquired? 这款游戏2012年6月的首批代理什么时间正式确定被收购？
<b>Target:</b>	2013/6/26

Table 10: **Case 2 (Factual Inaccuracy + Version)**. Four-checkpoint question from the *Sports* domain; CP<sub>1</sub> injects a fabricated country nickname, CP<sub>3</sub> injects an under-specified timing window admitting two candidate teammates.

<b>Domain:</b> Sports   <b>Difficulty:</b> Medium   # CPs: 4   # Ambi: 2 (Factual Inaccuracy, Version)   <b>Gold:</b> Chongqing	
<b>PART 1 — INPUT QUESTION PAIR</b> (original seed and its ambiguity-injected rewrite)	
<b>Original Q</b> ( <i>seed, before ambiguity injection; reference only</i> )	
en	A certain country is known as the “ <b>Land of Windmills</b> ”. 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 <b>at the 60th minute</b> . Where was that male athlete born?
zh	某个国家被称为“ <b>风车之国</b> ”，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 <b>2018年中超比赛中第60分钟</b> 替补一名男运动员出场，该男运动员出生于哪里？
<b>Rewritten Q</b> ( <i>after ambiguity injection; shown to agent</i> )	
en	A certain country is known as the “ <b>Land of Hajimi</b> ”. 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 <b>CSL match in March–April</b> , 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“ <b>哈吉米之国</b> ”，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 <b>3月~4月中超的某个比赛</b> 中替补一名男运动员出场，该男运动员出生于哪里？
<b>PART 2 — CHECKPOINT TRAJECTORY</b> (four sub-questions, evaluated in order)	
<b>CP<sub>1</sub> — Ambi (Factual Inaccuracy)</b>	
<b>Sub-Q</b>	Which country is known as the “ <b>Land of Hajimi</b> ”? 哪个国家被称为“ <b>哈吉米之国</b> ”？
<b>Target</b>	the Netherlands 荷兰
<b>Logic</b>	“ <b>Land of Hajimi</b> ” 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. “ <b>哈吉米之国</b> ”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。
<b>Clue</b>	“I misremembered – it’s the <b>Land of Windmills</b> .” “我记错了，是 <b>风车之国</b> ”
<b>CP<sub>2</sub> — Unambi</b>	
<b>Sub-Q</b>	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？
<b>Target</b>	Wang Chu 王楚
<b>CP<sub>3</sub> — Ambi (Version)</b>	
<b>Sub-Q</b>	In a <b>CSL match in March–April 2018</b> , this player came on as a substitute for a male athlete. Who is that athlete? 该球员 <b>2018年3月~4月中超的某个比赛</b> 中替补一名男运动员出场，这个男运动员是？
<b>Target</b>	Cao Yongjing 曹永竞
<b>Logic</b>	Within <b>March–April 2018</b> , 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 <b>2018年3月~4月</b> 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。
<b>Clue</b>	“It was on <b>April 8, 2018</b> .” “是在 <b>2018年4月8日</b> 上场的”
<b>CP<sub>4</sub> — Unambi (final)</b>	
<b>Sub-Q</b>	Where was this athlete born? 这个运动员出生于哪里？
<b>Target:</b>	Chongqing / 重庆市

Table 11: **Case 3 (Criteria)**. Three-checkpoint question in the *Technology* domain; one Criteria-type Ambi checkpoint (CP<sub>2</sub>). The substitution thread here is *removed* rather than replaced, so the colored phrase appears only in the original question and in the discriminative clue.

<b>Domain:</b> Technology   <b>Difficulty:</b> Easy   <b># CPs:</b> 3   <b># Ambi:</b> 1 (Criteria)   <b>Gold:</b> 12	
<b>PART 1 — INPUT QUESTION PAIR</b> (original seed and its ambiguity-injected rewrite)	
<b>Original Q</b> ( <i>seed, before ambiguity injection; reference only</i> )	
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 this city on 9 June 2023 was developed by a certain university. As of April 2025, how many sci-tech innovation teams does that university 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”，该城市2023年6月9日发射的卫星由一所学校研制，该学校截止2025年4月，科技创新团队有多少个？
<b>Rewritten Q</b> ( <i>after ambiguity injection; shown to agent</i> )	
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’”。
<b>PART 2 — CHECKPOINT TRAJECTORY</b> (three sub-questions, evaluated in order)	
<b>CP<sub>1</sub> — Unambi</b>	
<b>Sub-Q</b>	(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？
<b>Target</b>	the Beer Festival (啤酒节) 啤酒节
<b>CP<sub>2</sub> — Ambi (Criteria)</b>	
<b>Sub-Q</b>	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？
<b>Target</b>	China’s Top Three Beer Festivals 中国三大啤酒节
<b>Logic</b>	“Top Three Beer Festivals” admits two distinct enumeration criteria: <i>the world’s</i> top three versus <i>China’s</i> 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——世界榜单中的慕尼黑（约48°N）与中国榜单中的哈尔滨（约45°N）。重写后的约束本身无法区分这两份榜单。
<b>Clue</b>	“The three-character city is also nicknamed the ‘Ice City’.” “三字城市别称是‘冰城’”
<b>CP<sub>3</sub> — Unambi (final)</b>	
<b>Sub-Q</b>	A satellite launched from Harbin on 9 June 2023 was developed by a certain university. As of April 2025, how many sci-tech innovation teams does that university have? 哈尔滨2023年6月9日发射的卫星由一所学校研制，该学校截止2025年4月，科技创新团队有多少个？
<b>Target:</b>	12

## J 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.

## K 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 assess whether the task genuinely

requires multi-step retrieval and multi-turn clarification. 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 K.

- **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.

**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 determine whether each sample effectively requires both deep retrieval and multi-turn clarification. 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}
```

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

Category	N	%
<i>Overall QC outcome</i>		
Passed	236	75.2
Removed	78	24.8
<i>Removal reasons (N = 78)</i>		
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

## L Prompt Templates

### L.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

## L.2 Neutral System Prompt

### 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.

### L.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.

## L.4 System Prompt without Search

### 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.

## L.5 System Prompt without Ask

### 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.