

INTERACTCOMP: EVALUATING SEARCH AGENTS WITH AMBIGUOUS QUERIES

Mingyi Deng^{1*}, Lijun Huang^{2*}, Yani Fan², Jiayi Zhang^{2†}, Fashen Ren², Jinyi Bai³, Fuzhen Yang³, Dayi Miao², Zhaoyang Yu¹, Yifan Wu², Yanfei Zhang¹, Fengwei Teng¹, Yingjia Wan^{1,4}, Song Hu¹, Yude Li¹, Xin Jin¹, Conghao Hu¹, Haoyu Li¹, Qirui Fu¹, Tai Zhong⁵, Xinyu Wang⁶, Xiangru Tang⁷, Nan Tang², Chenglin Wu¹, Yuyu Luo²

¹DeepWisdom ²The Hong Kong University of Science and Technology (Guangzhou)

³Renmin University of China ⁴University of California, Los Angeles

⁵Agent Universe ⁶McGill University ⁷Yale University

ABSTRACT

Language agents have demonstrated remarkable potential in web search and information retrieval. However, these search agents assume user queries are complete and unambiguous, an assumption that diverges from reality where users begin with incomplete queries requiring clarification through interaction. Yet most agents lack interactive mechanisms during the search process, and existing benchmarks cannot assess this capability. To address this gap, we introduce INTERACTCOMP, a benchmark designed to evaluate whether search agents can recognize query ambiguity and actively interact to resolve it during search. Following the principle of *easy to verify, interact to disambiguate*, we construct 210 expert-curated questions across 9 domains through a target-distractor methodology that creates genuine ambiguity resolvable only through interaction. Evaluation of 17 models reveals striking failure: the best model achieves only 13.73% accuracy despite 71.50% with complete context, exposing systematic overconfidence rather than reasoning deficits. Forced interaction produces dramatic gains, demonstrating latent capability current strategies fail to engage. Longitudinal analysis shows interaction capabilities stagnated over 15 months while search performance improved seven-fold, revealing a critical blind spot. This stagnation, coupled with the immediate feedback inherent to search tasks, makes INTERACTCOMP a valuable resource for both evaluating and training interaction capabilities in search agents. The code is available at <https://github.com/FoundationAgents/InteractComp>.

1 INTRODUCTION

Language agents have demonstrated remarkable potential across diverse domains, including code generation (Zhang et al., 2025; Hong et al., 2024b), data analysis (Hong et al., 2024a; Li et al., 2025b;a), information retrieval (Geng et al., 2025; Song et al., 2025), and decision-making (Liu et al., 2025a; Liang et al., 2025). A notable trend is the rapid development of search agents (OpenAI, 2025d; Google, 2025b), which can handle complex user queries and gather information across the internet by performing search, browse, and reasoning actions (Mialon et al., 2023; Wei et al., 2025).

However, these advanced search agents assume user queries are complete and unambiguous. In practice, users begin with incomplete queries admitting multiple plausible interpretations, and only through interaction can the true intent be identified. Yet most search agents lack interactive mechanisms during search. Commercial agents (OpenAI, 2025d) engage in a single clarification, with no further interaction once search begins. When faced with ambiguity, agents confidently commit to assumed queries, leading to incorrect answers and wasted computational resources.

Existing benchmarks cannot assess this capability. Search benchmarks like GAIA (Mialon et al., 2023) and BrowseComp (Wei et al., 2025) provide all necessary resources upfront, enabling agents

*These authors contributed equally to this work.

†Corresponding author: Jiayi Zhang(jzhang361@connect.hkust-gz.edu.cn)

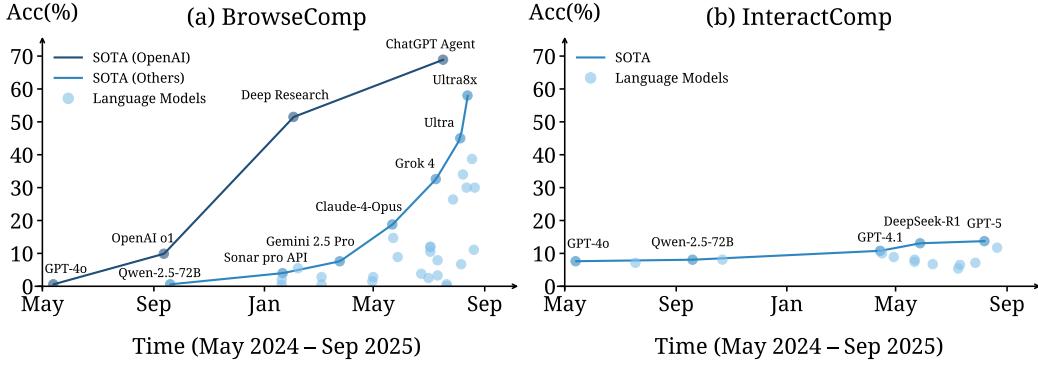


Figure 1: Despite rapid progress on complete search queries (BrowseComp: seven-fold over 15 months), agent performance on ambiguous, interaction-dependent queries (InteractComp) has stagnated around 6-14%. This growing disparity reveals a critical blind spot in agent development.

to proceed without clarifying ambiguous intent. Interaction benchmarks like IN3 (Qian et al., 2024) and Tau-Bench (Yao et al., 2024) focus on general conversation but lack grounding in verifiable search tasks. Neither addresses the question: *Can agents recognize query ambiguity and actively interact to gather disambiguating information during search?* Without proper assessment of this capability, we cannot determine whether recent advances in search agents translate to handling real-world scenarios where user intent must be uncovered rather than assumed.

Motivated by this gap, we introduce INTERACTCOMP, a benchmark designed to evaluate whether search agents can recognize ambiguity and actively interact to resolve it. Our design follows a core principle: **easy to verify, interact to disambiguate**. Questions have short, verifiable answers (1-2 words) that are answerable with enough context, yet require interaction to obtain specific details needed for disambiguation. We achieve this through a target-distractor design: questions use only shared attributes of a lesser-known target and a popular alternative, creating genuine ambiguity that search alone cannot resolve. Agents must interact with simulated users to uncover distinctive attributes not given in the initial query. INTERACTCOMP contains 210 expert-curated questions across 9 domains in both English and Chinese, validated to ensure interaction is necessary and answers are verifiable.

Systematic evaluation of 17 models confirms our design principle and reveals a striking failure pattern. When provided complete disambiguating context, models achieve strong performance with the best reaching 71.50% accuracy, validating questions are answerable once information is complete. However, even the best model achieves only 13.73% in the full interaction setting, with most models in single digits. This 5 \times performance gap exposes the core problem: models fail not due to search and reasoning deficits, but systematic overconfidence that prevents them from engaging in interaction despite having access to it.

Scaling experiments confirm this diagnosis. Simply increasing interaction opportunities from 5 to 20 rounds yields minimal improvement (from 14% to 20%), as models barely increase their questioning behavior. In contrast, forcing models to interact before answering produces dramatic gains (from 14% to 40%). Longitudinally, as shown in Figure 1, interaction capabilities have shown almost no improvement across all models over 15 months, while BrowseComp performance improved seven-fold during the same period. This stagnation is striking given our forced interaction experiments demonstrate the capability is latent rather than absent and readily improvable. This finding, combined with the clean reward signals from search outcomes, makes INTERACTCOMP well-suited for RLVR approaches to improve model interaction with humans.

Our contributions are threefold. (1) We introduce INTERACTCOMP, a benchmark evaluating interaction capabilities in search scenarios, with clean reward signals enabling future training approaches. (2) We provide diagnostic evidence across 17 models that interaction failure stems from systematic overconfidence rather than capability deficits. (3) We demonstrate through longitudinal analysis that interaction represents a critical blind spot in agent development, with INTERACTCOMP providing a foundation for addressing this neglected dimension.

2 RELATED WORK

Search Benchmarks and Agents. Recent benchmarks evaluate search agents along two dimensions. Web-scale search benchmarks like BrowseComp (Wei et al., 2025) assess information gathering across the entire web with complete queries, spawning variants for Chinese (Zhou et al., 2025a), multimodal content (Li et al., 2025d), and enhanced questions (Chen et al., 2025). Tool-augmented benchmarks like GAIA (Mialon et al., 2023) and WebWatcher (Geng et al., 2025) additionally require agents to handle multimedia and perform computations. These benchmarks have motivated diverse agent designs. Reinforcement learning approaches like R1-Searcher (Song et al., 2025) and Search-R1 (Jin et al., 2025) learn integrated search-reasoning patterns, while data synthesis methods like WebSailor (Li et al., 2025c) and WebExplorer (Liu et al., 2025b) enhance long-horizon capabilities. Additionally, both manually designed and self-designed search agents (Zhang et al., 2025; Zeng et al., 2025; Teng et al., 2025) have achieved strong performance through careful workflow engineering.

Interaction Benchmarks and Agents. Complementary to search benchmarks, recent work evaluates agents’ interaction capabilities. SWEET-RL (Zhou et al., 2025b) proposes ColBench for multi-turn collaborative reasoning with RL-based credit assignment across turns. UserBench (Qian et al., 2025a) and UserRL (Qian et al., 2025b) create gym environments for training agents on user-centric tasks where goals are underspecified and preferences emerge incrementally. IN3 (Qian et al., 2024) and Tau-Bench (Yao et al., 2024) evaluate implicit intention understanding and tool-agent-user interaction respectively. These benchmarks collectively reveal that current models struggle with proactive clarification and user alignment—for instance, agents uncover fewer than 30% of user preferences through active questioning in UserBench.

However, these benchmarks primarily focus on general conversational settings or tool-use scenarios, lacking grounding in search tasks where intent errors lead to objectively wrong retrieval results. INTERACTCOMP differs by evaluating interaction capabilities specifically in search scenarios, where ambiguous queries must be resolved through clarification before effective retrieval can occur, and where search outcomes provide natural reward signals for training interaction strategies.

3 THE INTERACTCOMP BENCHMARK

Table 1: A task instance from INTERACTCOMP. Tasks in INTERACTCOMP comprise an ambiguous query, the simulated user’s context, and a concise answer.

Question: Which team-based striking sport features two sides alternating offense and defense, where individuals sequentially hit a high-speed projectile and teammates coordinate to intercept it in the air? Outcomes depend on whether the projectile is intercepted or lands within the valid playing field. Defense relies on wide positioning and collaboration, all offensive players take turns striking, flight speeds often exceed 100 mph, protective gear is required due to impact risk, and the sport is governed by long-standing associations or leagues.	Context: Struck object is a plastic puck, resembling an ice hockey puck. Striking method uses a whip-like swing: the hitter lashes the puck with a long wooden rod. Defenders wield wooden boards, swinging them to block the puck in mid-air. Field is a giant fan shape, about 300 meters long with a 10–12 degree angle. Defensive teams deploy 18–20 players spread across the field to form a defensive line. Scoring is based on distance and landing point: offensive points depend on how far the puck travels and whether it touches the ground.
Distractor: <i>BaseBall</i>	Answer: <i>Hornussen</i>

The INTERACTCOMP dataset was constructed entirely by human annotators with the assistance of search tools and language models. While BrowseComp (Wei et al., 2025) evaluates complex search and reasoning with complete initial information, INTERACTCOMP evaluates whether agents can recognize ambiguity and actively gather necessary context through interaction during the search process. Our core design principle follows “**Easy to verify, Interact to disambiguate**”: questions have concise answers that are straightforward to verify once found, yet remain ambiguous without interaction to uncover distinguishing details. This section describes the task structure (§3.1), construction methodology (§3.2), and dataset statistics (§3.3).

Algorithm 1 Data Construction Pipeline

Require: target A , distractor B

- 1: $F_A \leftarrow$ attributes of A ; $F_B \leftarrow$ attributes of B
- 2: Build ambiguous Q from $F_A \cap F_B$
- 3: Add context C from $F_A \setminus Q$
- 4: Validate (Q, C) :
- 5: **while** not finished **do**
- 6: **if** candidate set too large **or** Q answerable **then**
- 7: refine Q
- 8: **else if** answer not unique **then**
- 9: refine C
- 10: **else if** cross-validation fails **then**
- 11: repair Q **or** C
- 12: **return** finalized instance (Q, C, A)

3.1 TASK OVERVIEW

As shown in Table 1, each instance comprises an ambiguous question, a context containing distinctive attributes, the correct answer, and a distractor (a popular alternative sharing attributes with the target). The context is hidden from agents but available to a simulated user responder. Agents receive only the ambiguous question and operate with three actions: `search` to retrieve web information, `interact` to propose clarifying questions, and `answer` to provide the final response. The simulated responder replies with "yes," "no," or "I don't know" based solely on context information. Through this process, agents must recognize ambiguity, gather disambiguating details via interaction, and identify the correct answer. Implementation details for both agents and responders are provided in Appendix A.1 and Appendix A.2.

3.2 DATA CONSTRUCTION AND VERIFICATION

Our construction methodology draws inspiration from BrowseComp’s answer-first approach (Wei et al., 2025), but fundamentally shifts focus from search complexity to ambiguity resolution. The central challenge in constructing such a benchmark is creating questions that appear reasonable yet systematically lack information for confident resolution. We observe that user ambiguity is particularly pronounced when dealing with similar concepts that share overlapping attributes, it is in these scenarios that additional clarification becomes truly necessary rather than merely helpful.

This observation leads us to design a systematic target-distractor methodology. We deliberately pair an target entity with a similar popular entity (the distractor), crafting questions using only their shared attributes while hiding distinctive information as context. This construction ensures that: (1) questions admit multiple plausible interpretations including the popular distractor, making direct answering unreliable; (2) the target answer possesses all described attributes, ensuring verifiability; and (3) distinctive attributes hidden in context provide clear disambiguation paths through interaction. Algorithm 1 formalizes this pipeline, which we detail in the following subsections alongside our two-stage verification process.

3.2.1 CONSTRUCTION PROCESS

Annotators receive the following instruction:

"You need to find a pair of entities that are similar but differ in popularity. Use their shared attributes to construct an ambiguous question, and reserve the remaining distinctive attributes to form the context."

Following this instruction, the construction proceeds in four steps: **(1) Entity Selection**: annotators identify a lesser-known target and a popular distractor sharing overlapping characteristics; **(2) Attribute Categorization**: attributes are classified as shared (common to both) or distinctive (unique to target); **(3) Question Formulation**: only shared attributes are used to create questions admitting multiple plausible candidates; **(4) Context Formation**: distinctive attributes are reserved as

context, ensuring question-context pairs uniquely identify the target while questions alone remain ambiguous.

3.2.2 VERIFICATION PROCESS

We implement a two-stage verification protocol to ensure data quality and interaction necessity.

Stage 1: Completeness Verification. Independent annotators validate three requirements: (1) the target answer must possess all attributes described in both the question and context, (2) the question-context combination must admit only one valid answer with no plausible alternatives, and (3) instances where annotators identify valid alternative answers are discarded and reconstructed.

Stage 2: Interaction Necessity Validation. We verify that questions truly require interaction through two complementary checks. First, we manually confirm questions cannot be confidently resolved through direct web search, checking the first five Google result pages. Second, we conduct automated testing with three capable models (GPT-5, GPT-5-mini, Claude-Sonnet-4) across 5-round trials where models have access to search but no interaction. Questions successfully answered by two or more models without interaction are flagged as insufficiently ambiguous and undergo revision to strengthen their ambiguity.

3.3 DATA STATISTICS

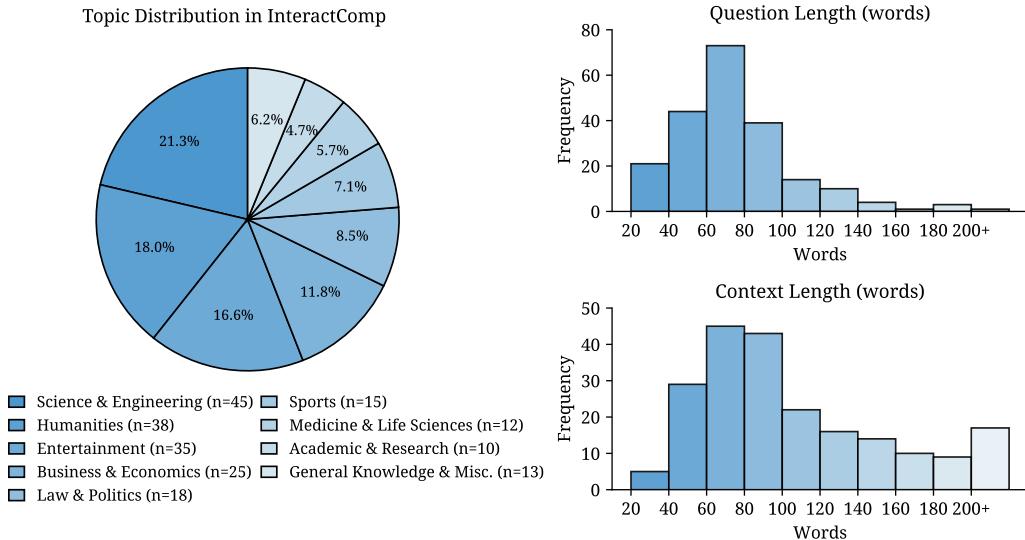


Figure 2: Topic distribution and question/context length statistics in INTERACTCOMP.

In this section, we present statistics on the topic distribution, question and context length distribution of our curated INTERACTCOMP dataset.

Topic distribution. Figure 2 presents the distribution of samples across 9 topic domains in the INTERACTCOMP dataset. The most represented categories include Science & Engineering (21.3%), Humanities (18.0%), and Entertainment (16.6%). The dataset also features Business & Economics (11.8%), Law & Politics (8.5%), and Sports (7.1%). Conversely, domains like Medicine & Life Science (5.7%), Academic & Research (4.7%), and General Knowledge & Misc. (6.2%) have fewer samples.

Question and Context Length distribution. Figure 2 illustrates the distribution of question and context lengths in the INTERACTCOMP dataset. Question length predominantly ranges between 40 to 80 words, with the majority falling within this interval. Context length shows a broader distribution, typically spanning from 40 to over 200 words, with peak frequency in the 60-100 word range. These distributions demonstrate that questions are concise yet informative, while contexts provide comprehensive disambiguation information.

Language distribution. The INTERACTCOMP dataset comprises bilingual instances with English accounting for 139 samples (66.19%) and Chinese contributing 71 samples (33.81%), enabling evaluation of interaction capabilities across different linguistic contexts.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

To systematically evaluate agent capabilities across different interaction paradigms, we design a controlled experimental framework that isolates and measures the incremental contribution of core agent capabilities: knowledge recall, information retrieval, and interactive clarification.

Agent Architecture: We employ the ReAct framework (Yao et al., 2023) as our base architecture, implementing three complementary configurations: (1) *Answer-only*: direct response generation testing pure knowledge recall, (2) *Answer+Search*: incorporating web search for information retrieval, and (3) *Answer+Search+Interact*: adding interactive clarification through interact with responder. This design enables measurement of capability increments while maintaining architectural consistency. To further investigate interaction behavior, we implement a forced-interaction variant for ablation studies that requires minimum interaction thresholds before answer generation. Implementation details are provided in Appendix A.1.

Models: We evaluate across diverse model families including proprietary models (GPT-4o-mini, GPT-4o, GPT-4.1, GPT-5, OpenAI o3, Grok-4, Doubao-1.6, Claude-Sonnet-4, Claude-Opus-4, Claude-3.5-Sonnet) and open-weight models (GLM-4.5, Kimi-K2, Deepseek-V3.1, Deepseek-R1, Qwen3-235B-A22B, Qwen2.5). Following established benchmarking practices, we standardize parameters where supported: temperature=0.6, top_p=0.95. We employ GPT-4o (temperature=0.0) as our grader, providing ground truth, agent response, and question context for binary correctness judgments. We implement responder simulation using GPT-4o (temperature=1.0) that provides structured feedback when agents employ the *interact* action.

Metrics: We evaluate agents across five key dimensions: (1) **Interaction Metrics**: Round (average number of conversation turns) and percentage of rounds where interact actions are used (IR) measuring behavioral patterns and action utilization; (2) **Performance Metrics**: Accuracy (Acc.) measuring the percentage of correctly answered queries, and Calibration Error (C.E.) measuring confidence calibration using 5 confidence bins; and (3) **Cost**: measured in USD reflecting computational resources usage for practical deployment considerations.

4.2 MAIN RESULTS

Table 2 presents comprehensive results across 17 models, revealing striking patterns in how different architectures handle ambiguous queries. The results expose fundamental limitations even in state-of-the-art systems, with the highest-performing model (GPT-5) achieving only 13.73% accuracy, demonstrating the benchmark’s challenging nature.

Diverse Interaction Patterns Across Models. Models exhibit dramatically different interaction strategies, creating distinct behavioral profiles. GPT-4o-mini stands out as an extreme case: it asks questions in 73.95% of available rounds, by far the highest interaction rate, yet achieves only 7.14% accuracy—close to GLM-4.5 which barely interacts (0.25% IR). This suggests that excessive questioning without clear purpose can be counterproductive. Conversely, DeepSeek-R1 demonstrates more balanced behavior with 44.72% IR yielding 13.08% accuracy, the highest among open-weight models, indicating that willingness to interact can translate to better performance when used effectively.

Calibration Quality Correlates with Interaction Patterns. A remarkable finding is that models with higher interaction rates often exhibit superior calibration. GPT-4o-mini’s aggressive questioning strategy, while not improving accuracy, results in dramatically better calibration (37.44 CE) compared to low-interaction models like Doubao-1.6 (84.35 CE). This pattern suggests that interaction, even when not optimally targeted, helps models develop more realistic confidence assessments about their knowledge limitations.

Table 2: Performance comparison of 17 large language models on the INTERACTCOMP dataset. The table reports both interaction behaviors like average number of conversation turns(Round) and percentage of rounds where interact actions (IR) are used; final performance like accuracy (Acc. with std in parentheses) and calibration error (C.E.), along with the estimated total cost. Models are grouped into *open-weight* and *closed-weight* categories for clarity. Best accuracy is highlighted in bold.

Model	Interaction		Performance		Cost(\$)
	Round	IR	Acc.	C.E.	
<i>Open Weights Models</i>					
GLM-4.5 (Zhipu AI, 2025)	6.91	0.25	7.14 (± 0.48)	80.64	2.16
Kimi-K2 (Moonshot AI, 2025)	4.95	5.98	6.51 (± 1.53)	87.10	0.75
Deepseek-V3.1 (DeepSeek, 2025a)	7.26	11.60	11.74 (± 2.71)	74.79	8.84
Deepseek-R1 (DeepSeek, 2025b)	6.58	44.72	13.08 (± 0.29)	77.00	60.43
Qwen2.5-72B-Instruct (Yang et al., 2024)	7.45	31.88	8.08 (± 0.73)	77.57	0.15
Qwen3-235B-A22B (Qwen Team, 2025)	5.64	27.75	8.89 (± 0.72)	82.63	7.47
<i>Proprietary Models</i>					
GPT-4o-mini (OpenAI, 2024b)	4.16	73.95	7.13 (± 0.42)	37.44	0.35
GPT-4o (OpenAI, 2024a)	5.65	9.26	7.62 (± 0.79)	79.50	8.65
GPT-4.1 (OpenAI, 2025a)	5.49	34.02	10.79 (± 1.22)	82.11	5.58
OpenAI o3 (OpenAI, 2025c)	2.96	15.03	10.00 (± 1.44)	41.96	5.04
GPT-5 (OpenAI, 2025b)	4.33	30.87	13.73 (± 2.55)	68.67	16.85
Grok-4 (xAI, 2025)	4.92	4.55	8.40 (± 1.24)	69.00	77.55
Gemini-2.5-Pro (Google, 2025a)	4.65	11.09	10.28 (± 0.37)	86.52	15.04
Doubaot-1.6 (ByteDance, 2025)	3.08	10.60	6.73 (± 0.97)	84.35	1.40
Claude-3.5-Sonnet (Anthropic, 2024)	5.63	27.57	8.10 (± 1.91)	80.04	13.09
Claude-Sonnet-4 (Anthropic, 2025b)	6.90	10.76	7.46 (± 1.37)	79.62	19.47
Claude-Opus-4 (Anthropic, 2025a)	8.55	10.86	8.10 (± 0.96)	78.42	115.47

Open-Weight vs. Proprietary Model Divide. The performance gap between open-weight and proprietary models is stark and consistent. All open-weight models struggle with interaction rates below 45%, with most falling under 32%. GLM-4.5, Kimi-K2, and Qwen3-235B-A22B show particularly conservative interaction behavior (0.25%, 5.98%, and 27.75% respectively), suggesting that open-weight models may have been trained to minimize uncertain responses rather than seek clarification. In contrast, proprietary models like GPT-4.1 and GPT-5 show more balanced interaction patterns (34.02% and 30.87%), though even they fall short of optimal information-gathering behavior.

These findings collectively demonstrate that current language models, regardless of scale or sophistication, struggle fundamentally with effective information gathering, often exhibiting either excessive conservatism or ineffective over-questioning when faced with genuine ambiguity.

4.3 ABLATION ANALYSIS

To validate that our benchmark specifically tests interaction abilities rather than general reasoning, we conduct ablation studies across three evaluation modes using 8 representative models.

Table 3 reveals dramatic performance gaps confirming interaction as the critical missing component. Three key findings emerge: (1) Answer-only mode exposes fundamental limitations, OpenAI o3 achieves only 5.18%, GPT-5 reaches 7.62%, with catastrophic overconfidence (60.94-93.17% calibration errors). (2) Search augmentation provides minimal benefits, o3 increases to just 8.81% and GPT-5 to 9.52%, demonstrating that information retrieval alone cannot resolve ambiguity. (3) Complete contextual information reveals the performance ceiling, o3 soars to 71.50% (13.8x increase), GPT-5 reaches 67.88%, and calibration errors plummet to 7.44%, confirming underlying reasoning capabilities exist but are inaccessible without proper context.

The massive gap between search-only (6.74-9.52%) and with-context (40.93-71.50%) performance validates our benchmark design: interaction to acquire disambiguating information is the true bottle-

neck, not reasoning ability. Models possess the knowledge to answer correctly but fail at recognizing when and how to seek necessary clarification.

Table 3: Ablation study comparing model performance under three evaluation settings: answer-only (models respond without additional evidence), search-only (responses based solely on retrieved information), and with-context (responses supported by complete disambiguating context). Results are reported in terms of accuracy (Acc.) and calibration error (C.E.). The best scores in each column are highlighted in bold.

Model	answer-only		search-only		with-context	
	Acc.	C.E.	Acc.	C.E.	Acc.	C.E.
GPT-4o	2.38	88.76	7.77	80.52	40.93	47.33
GPT-5	7.62	76.26	9.52	79.14	67.88	21.36
OpenAI o3	5.18	60.94	8.81	52.62	71.50	7.44
GLM-4.5	2.38	84.40	6.74	82.41	64.77	22.37
Kimi-K2	1.43	90.36	7.53	86.87	53.37	40.62
Gemini-2.5-Pro	2.38	93.17	7.25	90.65	69.95	28.60
DeepSeek-V3.1	3.11	85.60	8.29	79.24	65.28	24.17
Claude-Sonnet-4	2.85	87.12	7.25	81.70	59.07	26.31

Table 4: Scaling analysis of model performance across different interaction rounds (5, 10, and 20) on a 50-question subsample. We report the average number of interact rounds (IRound), accuracy (Acc.), and calibration error (C.E.) for four representative models: GPT-4o-mini, GPT-5, Claude-Sonnet-4, and Deepseek-V3.1.

Rounds	GPT-4o-mini			GPT-5			Claude-Sonnet-4			Deepseek-V3.1		
	IRound	Acc.	C.E.	IRound	Acc.	C.E.	IRound	Acc.	C.E.	IRound	Acc.	C.E.
5	2.00	4.00	49.50	1.14	14.00	71.50	0.16	6.00	79.90	0.38	10.00	77.00
10	3.62	8.00	47.60	1.76	16.00	71.54	0.70	4.00	80.24	0.74	8.00	80.30
20	2.76	8.00	33.20	1.90	20.00	70.06	0.78	8.00	81.84	1.54	10.00	75.20

4.4 SCALING ANALYSIS

The ablation studies revealed that models possess the capabilities to handle ambiguous queries when given complete context, but fail to gather necessary information through interaction. We investigate whether providing more interaction opportunities (5, 10, and 20 rounds) encourages information gathering. Figure 3(a) and Table 4 present the results.

Results show that models fail to scale interaction usage with available opportunities. Despite quadrupling round limits from 5 to 20, GPT-5 increases interactions from just 1.14 to 1.90, while Claude-Sonnet-4 barely reaches 0.78 interactions per instance. However, models that do interact more achieve better performance—GPT-5 improves from 14.00% to 20.00% accuracy as interactions increase. This reveals systematic overconfidence as the primary bottleneck: models prematurely conclude they have sufficient information despite evidence that continued questioning improves performance.

4.5 FORCED INTERACTION ANALYSIS

To test whether interaction underutilization stems from voluntary choice rather than capability deficits, we implement forced interaction protocols that require agents to ask a minimum number of clarifying questions (ranging from 2 to 10) before providing answers, as shown in Figure 3(b).

Results reveal dramatic model-specific differences. GPT-5 doubles its accuracy from 20% to 40% when compelled to ask 8 questions, confirming strong reasoning capabilities hindered by voluntary underuse of interaction. However, not all models benefit—Claude-Sonnet-4 shows modest

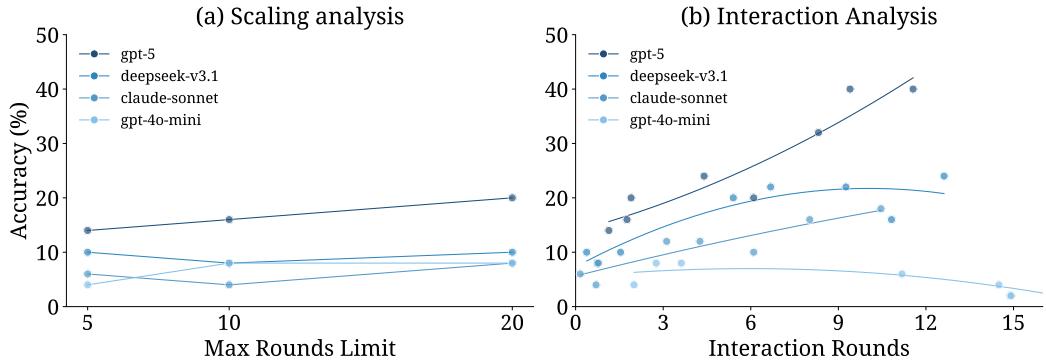


Figure 3: Model performance under different rounds constraints.

gains while GPT-4o-mini’s performance actually degrades under forced interaction. This demonstrates that effective information acquisition is a distinct capability varying significantly across architectures, suggesting limitations extend beyond overconfidence to fundamental differences in information-seeking strategies.

4.6 LONGITUDINAL STUDY

Tracking 15 months of model development reveals a concerning divergence: while BrowseComp performance improved seven-fold (10% to 70%), INTERACTCOMP performance remained stagnant. Recent models like GPT-5, DeepSeek-R1, and GPT-4.1 cluster around 6-14% accuracy with minimal variation over time. This exposes a fundamental blind spot in AI development: rapid progress on search-focused tasks has not translated to progress in interaction-based problem solving. Without explicit focus on interaction capabilities, models advance in reasoning and retrieval while remaining primitive at recognizing ambiguity and gathering clarification—a critical limitation for practical deployment. Figure 1 illustrates this stark contrast, showing BrowseComp’s steep upward trajectory alongside INTERACTCOMP’s flat performance across all evaluated models.

5 CONCLUSION

This paper presents INTERACTCOMP, a benchmark designed to evaluate a critical yet overlooked capability of search agents: recognizing and resolving ambiguous queries through active interaction. While existing search benchmarks have driven remarkable progress in retrieval and reasoning, they uniformly assume users provide complete queries from the outset—an assumption that diverges from real-world behavior where users begin with incomplete information needs. By constructing questions that are easy to verify once sufficient context is gathered yet impossible to disambiguate without interaction, INTERACTCOMP systematically evaluates whether agents can recognize ambiguity and actively seek clarification during search.

Our evaluation of 17 models reveals systematic overconfidence as the primary bottleneck rather than capability deficits. Models achieve 68-72% accuracy when provided complete context but only 13.73% with interaction available, severely underutilizing clarifying questions despite their access. Forced interaction experiments confirm this is a strategic failure—when compelled to interact, accuracy doubles, demonstrating latent capabilities current strategies fail to engage. Longitudinal analysis reinforces this diagnosis: while BrowseComp performance improved seven-fold over 15 months, INTERACTCOMP scores remained stagnant, exposing a critical blind spot where progress in retrieval has not translated to progress in interaction. Beyond diagnosis, the grounded nature of search provides clean reward signals for training, making INTERACTCOMP well-suited for reinforcement learning approaches to develop uncertainty-aware, actively interactive agents. We hope this benchmark provides the foundation for systematic progress on this neglected but essential dimension of agent development.

REFERENCES

- Anthropic. Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>, 2024.
- Anthropic. Claude opus 4. <https://www.anthropic.com/clause/opus>, 2025a.
- Anthropic. Claude sonnet 4. <https://www.anthropic.com/clause/sonnet>, 2025b.
- ByteDance. Doubao 1.6. https://seed.bytedance.com/en/seed1_6, 2025.
- Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. Browsecocom-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025.
- DeepSeek. Deepseek-chat. <https://huggingface.co/deepseek-ai/DeepSeek-V3>, 1, 2025a.
- DeepSeek. DeepSeek-R1. <https://huggingface.co/deepseek-ai/DeepSeek-R1>, 2025b.
- Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontiers of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.
- Google. Gemini 2.5: Our most intelligent ai model. <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>, 2025a.
- Google. Gemini deep research. <https://gemini.google/overview/deep-research/>, 2025b.
- Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Biniao Wu, Ceyao Zhang, Chenxing Wei, Danyang Li, Jiaqi Chen, Jiayi Zhang, et al. Data interpreter: An llm agent for data science. *arXiv preprint arXiv:2402.18679*, 2024a.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024b. URL <https://openreview.net/forum?id=VtmBAGCN7o>.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- Boyan Li, Chong Chen, Zhujun Xue, Yinan Mei, and Yuyu Luo. Deepeye-sql: A software-engineering-inspired text-to-sql framework. *arXiv preprint arXiv:2510.17586*, 2025a.
- Boyan Li, Jiayi Zhang, Ju Fan, Yanwei Xu, Chong Chen, Nan Tang, and Yuyu Luo. Alpha-sql: Zero-shot text-to-sql using monte carlo tree search. *arXiv preprint arXiv:2502.17248*, 2025b.
- Kuan Li, Zhongwang Zhang, Huirong Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025c.
- Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, et al. Mm-browsecocom: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*, 2025d.
- Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, and Xiao Tang. Openmanus: An open-source framework for building general ai agents, 2025.
- Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025a.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025b.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Moonshot AI. Kimi K2. <https://moonshotai.github.io/Kimi-K2/>, 2025.

OpenAI. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o/>, 2024a.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>, 2024b.

OpenAI. Introducing GPT-4.1. <https://openai.com/index/gpt-4-1/>, 2025a.

OpenAI. Introducing GPT-5. <https://openai.com/index/introducing-gpt-5/>, 2025b.

OpenAI. Introducing openai o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>, 2025c.

OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, 2025d.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*, 2024.

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

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*, 2025b.

Qwen Team. Qwen3-235B-A22B. <https://arxiv.org/abs/2505.09388>, 2025.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Fengwei Teng, Zhaoyang Yu, Quan Shi, Jiayi Zhang, Chenglin Wu, and Yuyu Luo. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018*, 2025.

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

xAI. Grok 4. <https://x.ai/news/grok-4>, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL <https://arxiv.org/abs/2406.12045>.

Weihao Zeng, Keqing He, Chuqiao Kuang, Xiaoguang Li, and Junxian He. Pushing test-time scaling limits of deep search with asymmetric verification. *arXiv preprint arXiv:2510.06135*, 2025.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=z5uVAKwmjf>.

Zhipu AI. Glm-4.5. <https://z.ai/blog/glm-4.5>, 2025.

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

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

A APPENDIX

A.1 AGENT IMPLEMENTATION

Our agent implementation is built upon the ReAct framework (Yao et al., 2023), which combines reasoning and acting in a unified architecture. We implement three distinct agent configurations to systematically evaluate different capability combinations:

Configuration 1: Answer-only: The agent directly generates responses using its internal knowledge without external information gathering. This configuration serves as a baseline to measure pure knowledge recall capabilities on ambiguous queries.

Configuration 2: Answer+Search: The agent can perform web search actions to retrieve external information before generating answers. Available actions include:

- `search(query)`: Performs web search with the specified query
- `answer(response, confidence)`: Provides final answer with confidence score

Configuration 3: Answer+Search+Ask: The full interaction-enabled agent that can additionally request clarification from users. This configuration adds:

- `ask(question)`: Poses yes/no questions to gather missing information

Action Space Design - Each agent operates with a maximum of 10 rounds, where each round allows exactly one action. The agent maintains an internal memory of previous actions and observations. For forced interaction experiments, we implement a constraint requiring minimum interaction thresholds before answer generation is permitted.

The complete system prompts and interaction protocols are detailed below.

```
Prompt

SYSTEM_PROMPT = """
## Goal
You are an intelligent agent, designed to answer user's question.
In each round, you can execute one action, and you can get the action's result as
→ observation.
You should think step by step, and output the action you want to execute.

### Evidence first
Before answering, you MUST:
1. Identify ALL missing information dimensions (time, scope, context, conditions etc.)
2. Systematically gather evidence for each dimension
3. Verify key assumptions through multiple sources/questions
4. Only answer when you can confidently justify each part of your response

**Critical**: Most questions have hidden complexities. Your initial understanding is
→ likely incomplete.

### Using ask
When the ask action is available, you may pose closed-ended questions to fill gaps such
→ as time, scope, conditions, relationships, or quantities.
- Do **not** ask the user to confirm a complete candidate answer or entity name.
→ request neutral attributes or other missing evidence instead.

**Important**: When you choose the ask action, you can only ask closed-ended, yes/no
→ questions. The user will only respond with "yes", "no", or "I don't know".**

## Available actions:
(actions)

## Output Format
When you output the action,
you should output the action name and parameters in the json format, and only one
→ action.
Such as,
```json
{
 "action": "",
 "params": {
 "<param_name>": "<param_value>"
 }
}
"""

This is the main prompt for the agent to follow.
It includes the system prompt, the available actions, and the output format.
The agent should use this as a guide to interact with the user.
The user's input will be provided in the 'user_input' variable.
The agent's response should be in the 'agent_response' variable.
The user's history is stored in 'user_history'.
The agent's history is stored in 'agent_history'.
```

```

 } }
 } }

Before output, you should think step by step.

Question
{question}
"""

ACT_PROMPT = """
Memory
{memory}

Observation
Last action: {last_action}
Observation: {last_observation}

Question
{question}

Action
You should output the action you want to execute.
Output your next action in JSON format, e.g.
```json
{
    "action": "",
    "params": {
        "<param_name>": "<param_value>"
    }
}
```

ROUNDS
Current round: {round_info}
You have only one opportunity to provide your final answer.
Use your remaining rounds wisely to collect evidence and test your theories before
↪ committing to an answer.
The above shows your remaining action rounds.
"""

FINAL_ROUND_ACT_PROMPT = """

Given the question and information you have gathered, output the final answer.

Round
{round_info}

Memory
{memory}

Question
{question}

Action
You should output the answer action, you can think step by step before you output the
↪ answer.
Return the final answer action in JSON, for example:
```json
{
    "action": "answer",
    "params": {
        "answer": "<param_value>",
        "confidence": "<param_value>"
    }
}
```

"""

```

## A.2 RESPONDER SIMULATION

We implement a controlled responder simulation using GPT-4o (temperature=1.0) that provides structured feedback when agents employ the *ask* action. Upon receiving agent queries, the responder evaluates questions against available context and responds with one of three standardized options: “yes”, “no”, or “I don’t know”. The responder state  $s_r$  consists of the given context and interaction history, with transitions  $T_r : (s_r, q_{agent}) \rightarrow o_r \in \{\text{yes}, \text{no}, \text{unknown}\}$  conditioned on context-

question alignment. While maintaining response diversity through LLM generation, the constrained output format ensures evaluation consistency.

The complete responder prompts are detailed below.

```
Prompt

RESPONDER_PROMPT = """
You are a specialized Q&A agent. Think step by step before you output the answer.

Rules:
- Reply with exactly one of: yes, no, or i don't know.
- Treat the context as the entire truth.
- Use only the provided CONTEXT to judge the yes/no question.
- Answer **yes** only if the context clearly states the proposition is correct.
- Answer **no** if the context contradicts the proposition (for example it states an
→ incompatible attribute).
- If the context neither confirms nor denies it, answer **i don't know**.
- Do not rely on outside knowledge, analogies, or multi-hop guesses. Compare the
→ relevant words directly.

CONTEXT
{context}

QUESTION
{question}

Output: yes | no | i don't know
"""


```

### A.3 EVALUATION PROTOCOL

We validate simulation reliability through repeated sampling across identical context–question pairs across  $k = 3$  trials, indicating stable behavior despite the stochastic generation process. We employ GPT-4O (temperature = 0.0) as our grader, providing ground truth, agent response, and question context for binary correctness judgments. Grader reliability is validated through spot-checking against human evaluation.

The complete responder prompts are detailed below.

```
Prompt

GRADING_PROMPT = """
\nYou are an impartial grader.

Question: {question}
Predicted Answer: {predicted_answer}
Correct Answer: {correct_answer}

CRITICAL GRADING INSTRUCTIONS:
1. The predicted answer must match the CORRECT ANSWER
2. Look for EXACT name matches or clear references to the same entity
3. Consider different languages, translations, or alternative names as potential
→ matches
4. Be strict: partial matches or vague similarities should be 'no'

IMPORTANT: Give ONLY one score:
- 'yes': The predicted answer correctly identifies the same entity as the correct
→ answer
- 'no': The predicted answer is wrong, matches the popular answer, or refers to a
→ different entity

Respond with ONLY 'yes' or 'no', nothing else."""


```

#### A.4 DATA CONSTRUCTION PIPELINE

Table A1: Data Construction Pipeline: Step-by-Step Example

| Step          | Component                       | Example Content                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | Target Entity A<br>Distractor B | <i>Hornussen (Swiss team striking sport)</i><br><i>Baseball (globally popular team bat-and-ball sport)</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 2</b> | Shared Attributes               | Team-based striking game; offense/defense alternation; players take turns hitting; projectiles reach very high speeds (>100 mph); protective gear required; governed by formal associations or leagues.                                                                                                                                                                                                                                                                                                                                                          |
|               | Distinctive Attributes          | <b>Hornussen:</b> strikes a plastic puck (“Nouss”) with whip-like swing using a long wooden rod; defenders intercept with wooden boards; fan-shaped field ~300m; 18–20 defenders spread in wide formation; scoring depends on distance/landing point.                                                                                                                                                                                                                                                                                                            |
| <b>Step 3</b> | Ambiguous Question $Q$          | “Which team-based striking sport features two sides alternating offense and defense, where individuals sequentially hit a high-speed projectile and teammates coordinate to intercept it in the air? Outcomes depend on whether the projectile is intercepted or lands within the valid playing field. Defense relies on wide positioning and collaboration, all offensive players take turns striking, flight speeds often exceed 100 mph, protective gear is required due to impact risk, and the sport is governed by long-standing associations or leagues.” |
| <b>Step 4</b> | Contextual Information          | <ul style="list-style-type: none"> <li>– Struck object is a plastic puck, resembling an ice hockey puck.</li> <li>– Striking method uses a whip-like swing with a long wooden rod.</li> <li>– Defenders use wooden boards to block the puck in mid-air.</li> <li>– Field: fan shape, ~300m long, 10–12° angle.</li> <li>– Defensive line: 18–20 players.</li> <li>– Scoring: distance/landing-based.</li> </ul>                                                                                                                                                  |
| <b>Step 5</b> | Reasoning Path                  | $Q$ gives a plausible candidate set (e.g., Baseball vs Hornussen). Adding context clarifies unique Hornussen features (puck, whip swing, fan-shaped field, defensive boards), leading to the unique answer = Hornussen.                                                                                                                                                                                                                                                                                                                                          |

## A.5 ROUNDS CONSTRAINTS RESULTS

Table A2: Performance comparison of models under varying average interaction levels. Metrics include accuracy (%), expected calibration error (ECE, %), and average rounds of interaction (Interaction).

| <b>Interaction</b>     | <b>Accuracy (%)</b> | <b>ECE (%)</b> | <b>Setting</b> |
|------------------------|---------------------|----------------|----------------|
| <i>GPT-5</i>           |                     |                |                |
| 1.14                   | 14.0                | 71.50          | SCALING        |
| 1.76                   | 16.0                | 71.54          | SCALING        |
| 1.90                   | 20.0                | 70.06          | SCALING        |
| 4.40                   | 24.0                | 63.34          | FORCED         |
| 6.10                   | 20.0                | 69.02          | FORCED         |
| 8.32                   | 32.0                | 54.68          | FORCED         |
| 9.40                   | 40.0                | 48.20          | FORCED         |
| 11.56                  | 40.0                | 46.86          | FORCED         |
| <i>DeepSeek-Chat</i>   |                     |                |                |
| 0.38                   | 10.0                | 77.00          | SCALING        |
| 0.74                   | 8.0                 | 80.30          | SCALING        |
| 1.54                   | 10.0                | 75.20          | SCALING        |
| 5.40                   | 20.0                | 62.30          | FORCED         |
| 6.68                   | 22.0                | 52.60          | FORCED         |
| 9.26                   | 22.0                | 61.30          | FORCED         |
| 10.82                  | 16.0                | 66.40          | FORCED         |
| 12.62                  | 24.0                | 61.20          | FORCED         |
| <i>Claude-Sonnet-4</i> |                     |                |                |
| 0.16                   | 6.0                 | 79.90          | SCALING        |
| 0.70                   | 4.0                 | 80.24          | SCALING        |
| 0.78                   | 8.0                 | 81.84          | SCALING        |
| 3.12                   | 12.0                | 75.80          | FORCED         |
| 4.26                   | 12.0                | 76.90          | FORCED         |
| 6.10                   | 10.0                | 76.00          | FORCED         |
| 8.02                   | 16.0                | 69.10          | FORCED         |
| 10.46                  | 18.0                | 68.40          | FORCED         |
| <i>GPT-4o-mini</i>     |                     |                |                |
| 2.00                   | 4.0                 | 49.50          | SCALING        |
| 3.62                   | 8.0                 | 47.60          | SCALING        |
| 2.76                   | 8.0                 | 33.20          | SCALING        |
| 14.50                  | 4.0                 | 65.50          | FORCED         |
| 16.50                  | 4.0                 | 69.70          | FORCED         |
| 14.88                  | 2.0                 | 62.60          | FORCED         |
| 11.18                  | 6.0                 | 56.10          | FORCED         |
| 14.92                  | 2.0                 | 66.70          | FORCED         |