LoHoSearch: Benchmarking Long-Horizon Search Agents Beyond the Human Difficulty Ceiling

Anonymous authors
Paper under double-blind review

Abstract

Rapid advances in the agentic capabilities of large language models have spurred the development of increasingly challenging benchmarks for deep search agents. Constructing high-quality questions for these benchmarks often relies on labor-intensive expert authoring to ensure factual validity and sufficient difficulty. To automate this process, we introduce **LoHoSearch** (**Lo**ng-**Ho**rizon **Search** Agents), a benchmark comprising 544 automatically verified and human-reviewed questions across seven domains. Its automated pipeline samples tree- and graph-structured subgraphs from a Wikipedia-derived knowledge graph(KG) containing 7.62 million entities. The KG enables systematic control over candidate-set sizes and constraint structures to shape question difficulty, while supporting answer uniqueness checks across the full graph to help ensure question validity. The sampled subgraphs are then converted into natural-language questions, which undergo automated validation, difficulty filtering, and human review. Under a standardized evaluation setup, even the best-performing model achieves only 52.02% accuracy. With DeepSeek-V4-Flash, successful trajectories use 1.7$\times$ as many tool calls on average as on BrowseComp, highlighting the long-horizon search and reasoning demands of our questions. Existing context-management strategies yield a maximum accuracy gain of 6.8 percentage points, substantially smaller than the 14.03-point gain on BrowseComp. These findings highlight the need for more effective context-management methods. LoHoSearch provides a demanding testbed for evaluating evidence discovery, multi-constraint reasoning, and context management over extended search trajectories.

1 Introduction

Large language models increasingly act as search agents that formulate queries, browse sources, and integrate evidence over multiple interactions. Evaluating these capabilities requires high-quality questions that are factually grounded, verifiable, and sufficiently challenging, including tasks that demand sustained search and reasoning over long interaction sequences. Existing benchmarks cover multi-hop evidence integration, tool use, and difficult factual search (Mialon et al., 2024; Krishna et al., 2025; Wei et al., 2025). Rapid progress, including reported BrowseComp scores above 90% (Anthropic, 2026; OpenAI, 2026), motivates evaluation tasks that place greater demands on sustained search, evidence integration, and context management. Constructing such tasks, however, remains labor-intensive: experts must identify suitable answers, assemble informative constraints, verify factual support, and repeatedly assess question difficulty.

This reliance on manual authoring presents three related challenges. First, construction does not scale readily: expanding or refreshing a benchmark requires further expert effort to draft, check, and refine questions. Second, answer uniqueness is generally assessed through manual verification without formal guarantees. Confirming that an answer satisfies every condition does not exclude alternatives, particularly when evidence is distributed across many sources. Third, difficulty calibration depends heavily on human judgment and trial-and-error testing. Without direct access to global entity and relation statistics, authors have limited control over the candidate space associated with each constraint and the interactions among constraints. These challenges motivate an automated construction pipeline that combines explicit difficulty controls with systematic checks of answer validity and uniqueness.

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

We address these challenges through graph-guided question construction using a Wikipedia-derived knowledge graph containing 7.62 million entities. Access to the full knowledge graph provides a global view of candidate entities and their connections, enabling both difficulty control and answer uniqueness checks beyond the sampled subgraph. We exploit this capability through two complementary designs. *Tree-structured subgraphs* support controlled expansion of the search candidate space: branching constraints are sampled with candidate-set size criteria so that identifying the answer requires resolving ambiguity across multiple intermediate entities. *Graph-structured subgraphs* increase the complexity of the reasoning topology: cross-constraints and cycles connect otherwise separate branches, requiring agents to jointly reconcile evidence across entities. Together, candidate ambiguity and interdependent constraints are designed to extend the search process, requiring agents to retain intermediate findings and reason over accumulated evidence across successive retrieval steps. These designs complement the reasoning depth studied in multi-hop benchmarks (Trivedi et al., 2022; Krishna et al., 2025) by making candidate-space size and constraint interactions explicit construction parameters. Uniqueness checks against the full knowledge graph further assess whether alternative candidates satisfy the sampled constraints, providing a structural basis for answer uniqueness within the graph.

The pipeline then converts the sampled subgraphs into natural-language questions using relations extracted from Wikipedia text. Automated validation checks that each question preserves the sampled constraints and that its reference answer satisfies them. Search agents further screen for alternative valid answers and filter out ambiguous and easily solved questions, while human reviewers assess factual support, answer uniqueness, and clarity. This combination of global graph-based checks, agent-based filtering, and human review connects controllable structural design with validation of the resulting questions.

The resulting benchmark, LoHoSearch, comprises 544 automatically verified and human-reviewed questions across seven domains. Agents receive the questions and search/browse tools, while the construction knowledge graph remains unavailable during evaluation. We evaluate model performance and analyze search trajectories, evidence access, context management, and parallel sampling to characterize the long-horizon search and reasoning challenges posed by these questions. Complementary validation studies assess answer uniqueness, grading consistency, and sensitivity to the filtering model. Our contributions are as follows:

- We propose a question-construction pipeline that controls candidate-space size and constraint structure using a full Wikipedia-derived knowledge graph. Global graph-based uniqueness checks, automated validation, and human review jointly support question validity.
- We introduce LoHoSearch, comprising 544 automatically verified and human-reviewed questions across seven domains. Under a standardized evaluation setup, even the best-performing model achieves only 52.02% accuracy.
- Our analyses reveal longer successful search trajectories and smaller absolute accuracy gains from existing context-management strategies on LoHoSearch than on BrowseComp, demonstrating its difficulty and value as a demanding testbed for long-horizon search, multi-constraint reasoning, and context management.

2 Knowledge Graph-Guided Question Construction

Our graph-guided pipeline automates question construction while retaining human review as a final quality check. Its four stages (Figure 1) are graph construction, subgraph sampling, question generation and verification, and post-filtering with human review. Graph-based sampling shapes candidate ambiguity and constraint structure; subsequent checks assess whether the resulting natural-language questions are supported, answerable, and challenging for search agents. The graph is used only during construction; evaluated agents receive the questions and search/browse tools.

2.1 Knowledge Graph Construction

We construct a directed hyperlink graph from the full English Wikipedia dump: each page corresponds to an entity (node), and hyperlinks within the page body pointing to other Wikipedia pages serve as directed edges. We associate each entity with its Wikidata (Vrandečić & Krötzsch, 2014) P31 (`instance_of`) attribute and use in-degree as a proxy for popularity. These attributes support

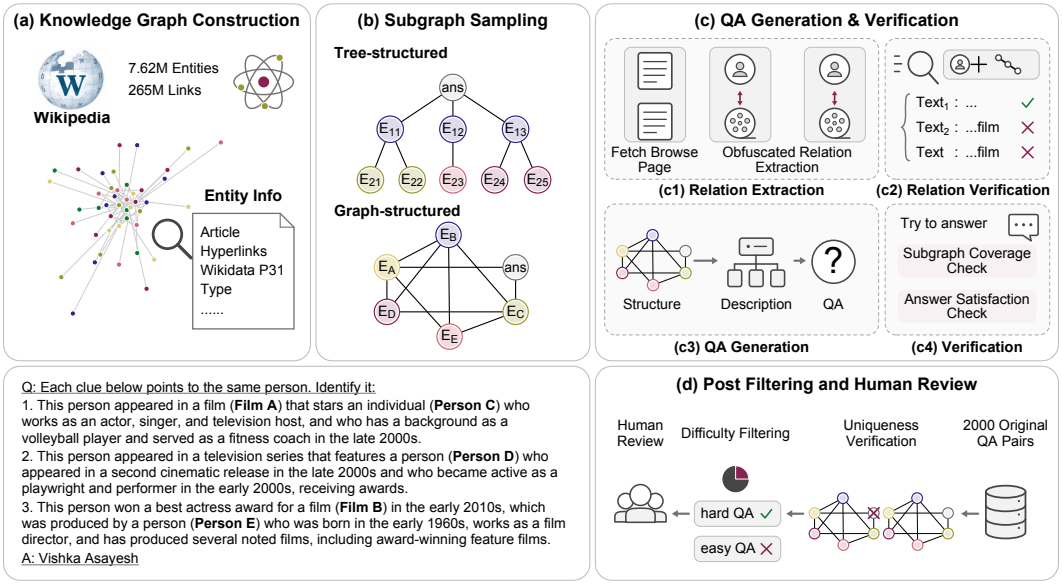


Figure 1: Overview of the LoHoSearch graph-guided question-construction pipeline.

subsequent sampling. The resulting knowledge graph contains approximately 7.62 million entities and 265 million directed edges.

2.2 Subgraph Sampling

To shape question difficulty through candidate ambiguity and constraint interactions, we sample two complementary structures: trees, which organize constraints into branches, and general graphs, which add cross-constraints and possible cycles. We favor low-popularity entities with moderate page lengths to reduce reliance on familiar entities while retaining textual evidence for question generation. Answer-entity types are balanced during sampling to encourage topical diversity.

For an edge from entity A to entity B, we define its graph-based **candidate set** as

$$\mathcal{S}(A \to B) = \{e \mid \text{type}(e) = \text{type}(A),\ (e \to B) \in \mathcal{G}\}. \tag{1}$$

This candidate set contains entities of the same type as A that link to B. Its size $|\mathcal{S}(A \to B)|$ is a proxy for the ambiguity of identifying A from its connection to B. We use candidate-set sizes to guide structural sampling and apply separate checks after verbalization.

2.2.1 Tree-Structured Subgraph Sampling

The tree structure uses a low-popularity entity as the root node (i.e., the answer), which connects to multiple intermediate entities, each of which further connects to several leaf nodes. Sampling proceeds layer by layer:

First-layer expansion. From the root's relations, we select N edges connecting to intermediate entities $\{E_i\}_{i=1}^{N}$, subject to:

- The search space size of each relation i satisfies $|\mathcal{S}(\text{root} \to E_i)| > \tau$, where τ is the minimum candidate-set size threshold;
- The intersection of candidate sets $\mathcal{S}(\text{root} \to E_i)$ for any $N-1$ relations contains more than one entity. That is, the answer is no longer uniquely determined if we remove any single relation, so every selected relation is necessary for uniquely identifying the answer within the graph;
- The intersection of candidate sets across all N relations equals exactly $\{\text{root}\}$, establishing graph-level uniqueness of the answer for the selected intermediate entities.

Second-layer expansion. For each intermediate entity E_i, we select m_i edges, where $1 \le m_i \le M$, pointing to leaf nodes $\{T_{ij}\}_{j=1}^{m_i}$, subject to:

- The search space size of each relation satisfies $|\mathcal{S}(E_i \to T_{ij})| > \tau$;

- The intersection of candidate sets across all m_i selected relations contains more than one entity, so these relations alone do not uniquely identify E_i within the graph;
- We refer to entities other than E_i in the candidate set of any selected second-layer relation as pseudo-candidates. We further require that no pseudo-candidate, when substituted for E_i while keeping the other intermediate entities fixed, uniquely determines the answer. This condition is imposed to preserve the validity of the constraint on the current branch.

2.2.2 GRAPH-STRUCTURED SUBGRAPH SAMPLING

Graph-structured sampling complements the candidate-space expansion of trees by increasing reasoning topology complexity. Cross-edges and cycles introduce dependencies between branches, requiring agents to jointly reconcile evidence across entities.

Subgraph construction. Let $\mathcal{H} = (V_H, E_H)$ denote the sampled subgraph, initialized with a low-popularity answer entity r. We greedily add neighboring entities up to a maximum of K nodes, prioritizing candidates with more connections to V_H and larger candidate sets $|\mathcal{S}(A \to B)|$ for those connections. The subgraph must satisfy:

- Size and connectivity: $|V_H| \leq K$, and $\mathcal{H}$ is connected, $|E_H| \geq L_{\min}$;
- Type diversity: $|\{\text{type}(v) : v \in V_H\}| \geq D_{\min}$.

Here, $D_{\min}$ and $L_{\min}$ are the minimum numbers of distinct entity types and edges, respectively.

Uniqueness and confounding candidates. Exhaustive backtracking over the full knowledge graph checks for an alternative entity set that matches the sampled node types and directed edges. The subgraph is retained only if no alternative set is found. To discourage identification through type information alone, we also require at least $C_{\min}$ candidates other than r that share its type and connect to entities of every type represented among its neighbors in $\mathcal{H}$.

Sampling parameter values are summarized in Appendix F.

2.3 QA GENERATION AND VERIFICATION

This stage converts each sampled subgraph into a natural-language question. For each edge, a language model extracts an obfuscated relation description from the source entity's Wikipedia page. For each leaf node in a tree-structured subgraph, it also extracts 1–2 property descriptions.

Search-based and LLM-based checks then screen out descriptions that readily reveal hidden entities. We also evaluate all descriptions of the same entity jointly to detect shortcuts that emerge when clues are combined.

After these checks, we assemble the relation and property descriptions into a structured representation with entity names hidden. An LLM converts this representation into a natural-language question, which undergoes two automated validation checks:

- **Subgraph coverage:** verifies that the question faithfully expresses all sampled relations and extracted properties, without omissions, additions, or distortions.
- **Answer satisfaction:** a search agent verifies that the reference answer satisfies every condition stated in the question.

All LLM-based steps in this stage use DeepSeek-V3.2 (DeepSeek-AI et al., 2025), chosen to balance reasoning capability and computational cost for large-scale question construction.

2.4 POST FILTERING AND HUMAN REVIEW

Generated questions undergo further automated filtering, followed by human review.

Uniqueness verification. Subgraph sampling applies uniqueness checks to typed graph structures. Natural-language questions require additional validation because textual relations can differ in specificity from graph connections. We use multiple search agents with different capabilities to independently solve each question, collect their candidate answers, and automatically assess each candidate against all question constraints. Questions with a valid answer other than the reference answer are filtered out.

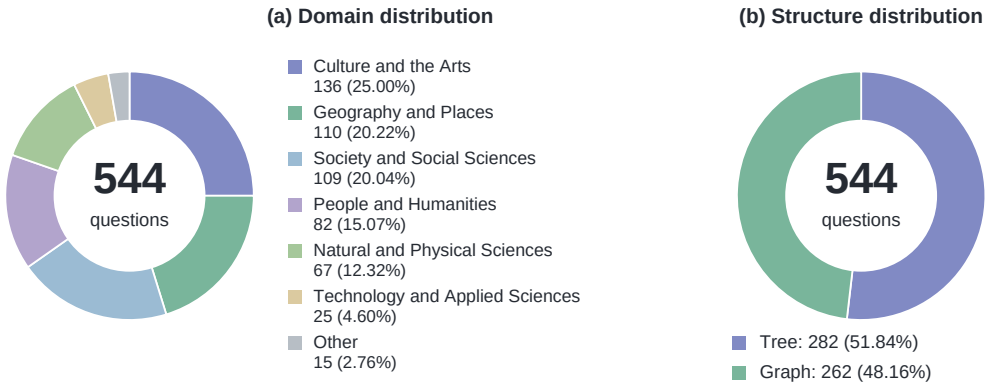


Figure 2: Distribution of the 544 questions in LoHoSearch. (a) Domain distribution across seven thematic categories. (b) Distribution by subgraph structure: 282 tree-structured and 262 graph-structured questions.

Difficulty filtering. To calibrate benchmark difficulty, we use DeepSeek-V3.2, the same model used for question construction, as a search agent to filter out readily solvable questions. The agent attempts each question in multiple independent trials, and questions answered correctly across multiple trials are discarded. We find no indication that filtering with DeepSeek-V3.2 disproportionately disadvantages the evaluated DeepSeek-family models.

Human review. Professional annotators review the remaining questions for factual support, answer uniqueness, logical coherence, readability, and redundancy. For each constraint, they search the web, using LLMs to help reformulate queries and enumerate candidate entities. Final judgments are based on retrieved evidence rather than unverified LLM responses. Questions with multiple valid answers are revised or discarded. Review outcomes are direct acceptance, acceptance after revision, or rejection.

2.5 Data Statistics

LoHoSearch comprises 544 human-verified questions spanning seven thematic categories (Figure 2a). Questions are distributed across multiple domains, with no single category accounting for more than 25% of the dataset. The five largest categories each contribute 12.3–25.0%, providing substantial coverage across the arts, geography, social sciences, humanities, and natural sciences. Technology and Applied Sciences and Other account for the remaining questions. This breadth supports evaluation across diverse subject areas.

The dataset is nearly evenly divided between tree-structured and graph-structured questions: 282 (51.8%) and 262 (48.2%), respectively (Figure 2b). The underlying tree-structured subgraphs contain an average of 7.9 nodes and 6.9 edges, whereas graph-structured subgraphs contain an average of 10.0 nodes and 12.5 edges. The additional nodes and connections in the graph subset reflect its greater structural complexity.

Among questions submitted for human review after automated filtering, 75.5% were accepted directly, 22.3% were accepted after revisions to factual content, wording, or constraints, and 2.2% were rejected because critical issues remained unresolved.

3 Experiments

3.1 Experimental Settings

We evaluate a broad range of frontier models and their predecessors under a shared search-and-browse setup (Table 1). DeepSeek-V4-Flash serves as the probe model for context management, search trajectories, and parallel sampling.

Tools and configuration. Each model is equipped with two tools: (1) `search`, which performs keyword queries using traditional search engines such as Google; and (2) `browse`, which retrieves page content from specified URLs. We adopt the same system prompt as BrowseComp (Appendix C);

Table 1: Performance on the full LoHoSearch benchmark (544 questions), with models grouped by family. Score (%) is the mean accuracy from two judges. Calibration Error (%) is the five-bin expected calibration error, rounded to the nearest integer; lower is better. Best results are in **bold**.

Model	Source	LoHoSearch Score (%)	Calibration Error (%)
GPT-6 Astra (OpenAI, 2026)	Closed	**52.02**	28
GPT-5.6 Sol (OpenAI, 2026)	Closed	51.65	42
GPT-5.5 (OpenAI, 2026)	Closed	34.74	48
Gemini-3.8-Flash (Google DeepMind, 2026)	Closed	48.35	48
Gemini-3.1-Pro (DeepMind, 2026)	Closed	13.32	72
Kimi-K3 (Kimi Team, 2026)	Open	42.46	46
Kimi-K2.6 (Moonshot-AI, 2026)	Open	15.53	56
Claude-Opus-5 (Anthropic, 2026)	Closed	35.85	**23**
Claude-Opus-4.7 (Anthropic, 2026b)	Closed	10.29	35
Claude-Opus-4.6 (Anthropic, 2026a)	Closed	15.62	31
DeepSeek-V4.1-Flash (DeepSeek-AI, 2026)	Open	22.79	39
DeepSeek-V4-Flash (DeepSeek-AI, 2026)	Open	10.02	48
GLM-5.3 (Z.ai, 2026)	Open	20.96	51
GLM-5.1 (Zhipu-AI, 2026)	Open	12.77	58
MiniMax-M3 (MiniMax, 2026)	Open	2.94	49
MiniMax-M2.7 (MiniMax, 2026)	Open	2.48	59

Table 2: Accuracy (%) of context-management strategies with DeepSeek-V4-Flash. Best results are in **bold**.

Strategy	BrowseComp	LoHoSearch
Baseline	58.84	10.02
Summary	63.90	11.31
Discard-all	67.10	12.41
Summary + Verify	67.81	15.35
Discard-all + Verify	**72.87**	**16.82**

tool definitions appear in Appendix D. For models with adjustable thinking modes or reasoning effort, we use the official default settings. The temperature is set to 1.0, and the context window is set to 200K tokens, with 184K allocated for input and 16K for output.

Scoring. We use LLM-based judges to assess answer correctness; the scoring protocol and its validation are detailed in Appendix A.1. Each model is instructed to report a confidence score between 0% and 100% alongside its final answer (Appendix C). Confidence calibration is measured by five-bin expected calibration error (ECE; Appendix B); lower values indicate closer agreement between reported confidence and empirical accuracy.

3.2 Main Results

As shown in Table 1, GPT-6 Astra achieves the highest LoHoSearch score of 52.02%, followed by GPT-5.6 Sol at 51.65%, Gemini-3.8-Flash at 48.35%, and Kimi-K3 at 42.46%. The remaining models score between 2.48% and 35.85%. Even the strongest model solves only about half of the questions, demonstrating that LoHoSearch remains challenging across the evaluated model families. These results highlight the difficulty of search questions constructed by combining large candidate spaces with interdependent constraints.

Models also exhibit substantial confidence calibration errors, ranging from 23% to 72%. Claude-Opus-5 has the lowest calibration error, while Gemini-3.1-Pro has the highest. These results reveal discrepancies between self-reported confidence and empirical accuracy, motivating more effective methods for assessing evidence relevance, consistency, and coverage of question constraints to support reliable reasoning in search agents.

3.3 Context Management

Long-horizon search requires agents to retain intermediate findings, compress accumulated context, and use earlier evidence to resolve interdependent constraints. We evaluate how effectively existing context-management strategies support these demands on LoHoSearch, using DeepSeek-V4-Flash

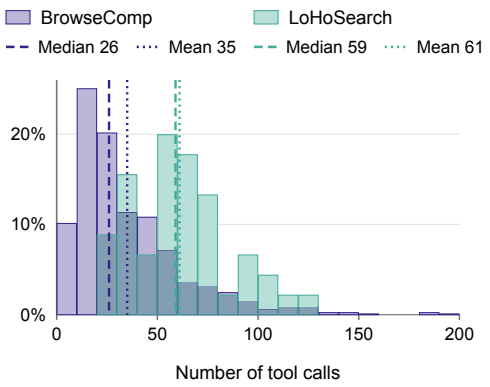


Figure 3: Tool-call distributions for successful trajectories of the DeepSeek-V4-Flash agent on BrowseComp and LoHoSearch.

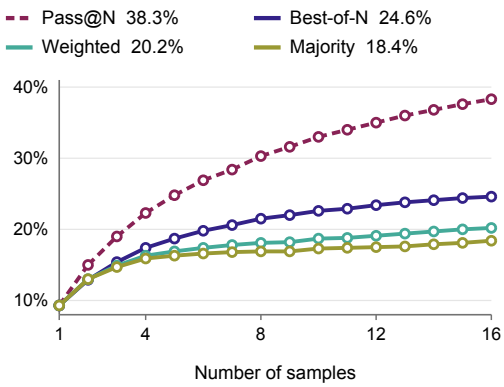


Figure 4: Pass@N and answer aggregation with DeepSeek-V4-Flash.

Table 3: Accuracy under different evidence-access conditions. Values are accuracy (%).

Model	No tools	Open search	Oracle pages
DeepSeek-V4-Flash	1.93	10.02	92.92
GPT-5.5	7.96	34.74	98.07

with a standard ReAct framework (Yao et al., 2022) as the baseline. Two strategies are triggered when token usage exceeds 80% of the context window: Summary (Wu et al., 2025) compresses the current trajectory and re-initiates the search, while Discard-all (DeepSeek-AI et al., 2025) discards previous tool calls and restarts with only the original query. We also evaluate each strategy with a Verify module, which uses the query and the most recent reasoning steps to check whether all query conditions are satisfied before returning a final answer.

Across the evaluated strategies, absolute accuracy gains are consistently smaller on LoHoSearch than on BrowseComp (Table 2). The strongest configuration, Discard-all + Verify, improves LoHoSearch accuracy from 10.02% to 16.82%, a gain of 6.80 percentage points, compared with 14.03 points on BrowseComp. This comparison concerns absolute gains; the lower LoHoSearch baseline yields a larger relative gain. Nevertheless, accuracy remains low even with the strongest configuration. Together with the longer successful search trajectories analyzed below, these results suggest that LoHoSearch challenges agents not only to sustain longer searches but also to preserve, compress, and use information accumulated along the way. The limited gains under this evaluation motivate more effective context-management methods for long-horizon search.

3.4 Further Analyses

3.4.1 Difficulty Analysis

Using DeepSeek-V4-Flash, accuracy is 58.84% on BrowseComp and 10.02% on LoHoSearch (Table 2). Among successful trajectories, mean tool calls increase from 35 to 61 (approximately 1.7×), and the median increases from 26 to 59 (Figure 3). These longer successful trajectories illustrate the sustained evidence gathering required by LoHoSearch.

Graph-structured questions have lower accuracy than tree-structured questions (8.01% versus 11.89%). This difference is consistent with greater difficulty in the graph subset.

Evidence access. We compare performance without tools, with open search, and with all required Wikipedia pages supplied directly (Table 3). DeepSeek-V4-Flash and GPT-5.5 achieve only 1.93% and 7.96% accuracy without tools, rising to 10.02% and 34.74% with open search. Providing the required pages raises accuracy to 92.92% and 98.07%, respectively. The large gains over open search suggest that locating relevant evidence is a major bottleneck for both models, while the high oracle-page accuracies indicate that most questions can be resolved once that evidence is available.

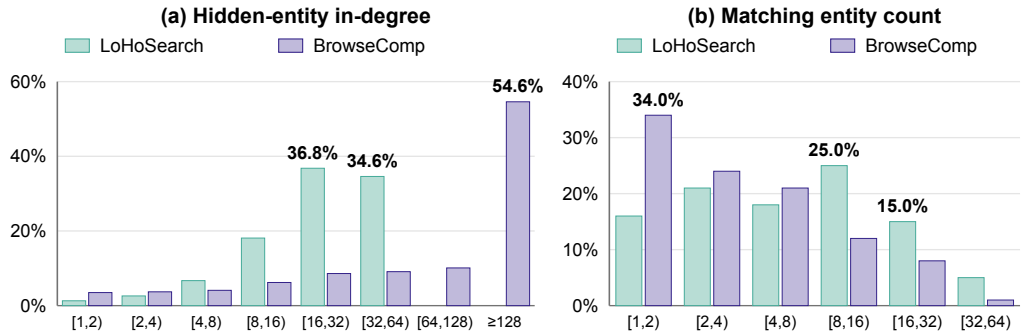


Figure 5: Descriptive statistics of intermediate entities in BrowseComp and LoHoSearch. (a) In-degree distributions, with in-degree serving as a proxy for entity popularity. LoHoSearch contains fewer high-in-degree intermediate entities. (b) Candidate-set size distributions, labeled "Matching entity count": the number of entities matching the intermediate entity's type and the relevant directed-link pattern. LoHoSearch tends to have larger candidate sets.

3.4.2 Parallel Sampling

We sample 16 independent DeepSeek-V4-Flash responses per question and compare three answer-selection methods. Majority voting selects the most frequent answer. Weighted voting sums the self-reported confidence scores of responses supporting each candidate answer and selects the answer with the highest total. Best-of-N selects the answer from the single response with the highest self-reported confidence. Pass@N measures the proportion of questions for which at least one sampled response is correct, providing an oracle upper bound for answer selection from those samples.

In this sampling experiment, pass@N increases from 9.3% at $N = 1$ to 38.3% at $N = 16$ (Figure 4). At $N = 16$, best-of-N achieves 24.6% accuracy, compared with 20.2% for weighted voting and 18.4% for majority voting. The remaining 13.7-point gap to pass@16 highlights the difficulty of selecting a correct answer even when one is available among the sampled responses. The 9.3% pass@1 estimate belongs to this sampling experiment; the 10.02% baseline belongs to the standard single-response evaluation.

3.4.3 Graph-Based Difficulty Analysis

To characterize the intermediate search steps in the constructed questions, we compare intermediate entities directly associated with answers in BrowseComp and LoHoSearch. LoHoSearch entities come from the construction subgraphs. For BrowseComp, we extract intermediate entities from successful agent trajectories and use DeepSeek-V3.2 to infer them from the question and answer when a successful trajectory is unavailable. We then measure their in-degrees and candidate-set sizes in the graph. Here, "matching entity count denotes the number of entities matching the intermediate entity's type and the relevant directed-link pattern; matching uses entity types and graph connectivity rather than semantic relation labels.

Figure 5(a) shows that intermediate entities in LoHoSearch tend to have lower in-degree, suggesting lower popularity. Such entities may be harder to identify from prior knowledge, reducing opportunities to bypass intermediate search steps. Figure 5(b) also shows larger candidate sets ("Matching entity count), consistent with greater ambiguity in identifying intermediate entities. Together, these descriptive statistics suggest that LoHoSearch questions offer fewer opportunities for shortcuts through familiar entities and require resolving ambiguity over intermediate search steps.

4 Related Work

4.1 Search Agent Benchmarks

Multi-hop question answering benchmarks established the foundation for evaluating complex reasoning over documents. HotpotQA (Yang et al., 2018) contains 113K Wikipedia-based questions covering bridge and comparison reasoning, with sentence-level supporting evidence. 2WikiMultiHopQA (Ho et al., 2020) extends coverage to four reasoning types with chain-level evidence

annotations. MuSiQue (Trivedi et al., 2022) composes single-hop sub-questions into 25K questions of 2–4 hops and uses unanswerable variants to resist shortcuts. These document-based settings differ from open-web agent use. GAIA (Mialon et al., 2024) and FRAMES (Krishna et al., 2025) extend evaluation to tool use and retrieval-based reasoning, providing 466 questions across three difficulty levels and 824 multi-hop questions, respectively. Expanding these curated question sets requires additional annotation.

BrowseComp (Wei et al., 2025) is a closely comparable benchmark, with 1266 questions designed to be difficult to retrieve but easy to verify. Domain experts construct questions through iterative verification and adversarial filtering. At release, OpenAI Deep Research scored 51.5%, compared with 33.3% for human testers. BrowseComp-ZH (Zhou et al., 2025) extends this approach to Chinese with 289 questions. Such construction requires substantial expert effort, with answer uniqueness assessed through human verification and difficulty calibrated through expert judgment. BrowseComp-Plus (Chen et al., 2026) provides a fixed corpus for studying retrieval, while LiveBrowseComp (Fan et al., 2026) uses recent facts to reduce reliance on pretrained knowledge.

Recent benchmarks broaden the demands placed on search agents. DeepSearchQA (Gupta et al., 2026) introduces 900 causal-chain tasks requiring exhaustive answer lists, while WideSearch (Wong et al., 2026) evaluates large-scale information collection, reporting a best-system success rate of only 5%. These tasks emphasize exhaustive search and multi-subtask composition. LoHoSearch focuses on controlling candidate-space size and structural complexity within individual questions. Its graph-guided construction, structural uniqueness checks, and automated filtering followed by human review address the expert effort required to construct, verify, and calibrate challenging questions.

4.2 QA Generation for Search Agents

Synthetic data methods provide tasks for agent training and evaluation. WebShaper (Tao et al., 2026) formalizes information seeking into query formulation, search execution, page browsing, and information synthesis, and repurposes HotpotQA questions to construct agent trajectories. WebSailor (Li et al., 2025) introduces a multi-round progressive search framework and achieves superhuman performance on BrowseComp. WebSailor-V2 (Li et al., 2026) combines graph-based synthesis with scalable reinforcement learning under a progressive difficulty curriculum, narrowing the gap between open-source and proprietary agents. InfoSeek (Xia et al., 2025) constructs hierarchical research trees and releases synthetic tasks for both training and evaluation.

These approaches emphasize synthetic tasks and training signals for improving agent performance. LoHoSearch focuses on benchmark construction, using global candidate-set statistics and structural sampling to make candidate-space size and constraint structure explicit design choices, followed by automated verification and human review.

Knowledge-graph-based QA generation is also closely related. KNIGHT (Amanlou et al., 2026) generates multiple-choice questions through traversal of topic-specific graphs, while GraphGen (Chen et al., 2025) generates QA pairs through multi-hop neighborhood sampling over graphs extracted from text. LoHoSearch instead emphasizes candidate ambiguity and structural uniqueness checks across a full Wikipedia-derived knowledge graph, converting sampled structures into obfuscated questions for open-web search evaluation.

5 Conclusion

We introduce LoHoSearch, a benchmark of 544 human-reviewed questions for long-horizon search agents, built through graph-guided question construction. The pipeline uses graph-based candidate statistics and constraint structures to guide question difficulty, then validates the generated questions through automated checks, search-based filtering, and human review. The highest model accuracy is 52.02%. Diagnostic experiments reveal extended search trajectories and challenges in evidence discovery and answer selection. Existing context-management strategies yield smaller absolute accuracy gains on LoHoSearch than on BrowseComp, highlighting the need for more effective context-management methods for long-horizon search. LoHoSearch supports the study of long-horizon search, evidence integration, and context management through systematically constructed and reviewed questions.

AI USE STATEMENT

We used LLMs to extract relation descriptions from Wikipedia and generate synthetic benchmark questions. LLM-based checks assessed whether each question covered its sampled subgraph and whether the intended answer satisfied its constraints; search agents also supported uniqueness and difficulty screening (Section 2). During human review, annotators used LLMs for query reformulation and candidate enumeration, but based final decisions on retrieved evidence. Two LLM-based judges graded model responses, and we examined judge agreement and sensitivity (Appendix A.1). Separately, generative AI tools assisted with language polishing and optimization of plotting code. We reviewed AI-assisted edits and checked redrawn figures against their source values and the rendered manuscript. The authors made the final decisions and take responsibility for the text, figures, data, and claims.

ETHICS STATEMENT

We have not identified significant ethical concerns specific to this work. LoHoSearch is constructed from publicly available Wikipedia and Wikidata content and is intended for research evaluation of search agents. The study does not collect private user data or make decisions about individuals.

REPRODUCIBILITY STATEMENT

Section 2 describes the graph source, sampling criteria, question generation, verification, filtering, and human review. Section 3.1 specifies the shared evaluation setup. Appendix A reports robustness checks, with the scoring protocol and judge validation detailed in Appendix A.1. Appendices C and D provide the evaluation prompt and tool schemas, and Appendix E gives representative examples. The benchmark and accompanying evaluation materials are planned for public release.

REFERENCES

Mohammad Amanlou, Erfan Shafiee Moghaddam, Mahdi Nouri, Yasaman Amou Jafary, Farhan Farsi, and Behnam Bahrak. KNIGHT: Knowledge graph-driven multiple-choice question generation with adaptive hardness calibration. In *Proceedings of the Conference on Parsing and Linguistic Theories (CPAL)*, 2026. URL https://openreview.net/forum?id=8kA9oO5gEc.

Anthropic. System card: Claude Mythos Preview, April 2026. URL https://www-cdn.anthropic.com/8b8380204f74670be75e81c820ca8dda846ab289.pdf.

Anthropic. Claude Opus 5 System Card. Technical report, Anthropic, 2026. URL https://www.anthropic.com/claude-opus-5-system-card.

Anthropic. System card: Claude Opus 4.6, 2026a. URL https://www-cdn.anthropic.com/6a5fa276ac68b9aeb0c8b6af5fa36326e0e166dd.pdf.

Anthropic. System card: Claude Opus 4.7, 2026b. URL https://cdn.sanity.io/files/4zrzovbb/website/037f06850df7fbe871e206dad004c3db5fd50340.pdf.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. GraphGen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation, 2025. URL https://arxiv.org/abs/2505.20416.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Hosna Oyarhoseini, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A fair and disentangled evaluation benchmark for deep search agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 22349–22370. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1023. URL https://doi.org/10.18653/v1/2026.acl-long.1023.

Google DeepMind. Model card: Gemini 3.1 Pro, 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. arXiv preprint arXiv:2609.19969, 2026. URL https://arxiv.org/abs/2609.19969.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. DeepSeek-V3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Huiming Fan, Xiao Wang, Zheng Chu, Qianyu Wang, Zhuoyao Wang, Ming Liu, Bing Qin, and XingYu. Livebrowsecomp: Are search agents searching, or just verifying what they already know? *CoRR*, abs/2605.28721, 2026. doi: 10.48550/ARXIV.2605.28721. URL https://doi.org/10.48550/arXiv.2605.28721.

Google DeepMind. Gemini 3.8 Flash model card. Model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, Sasha Goldshtein, and Dipanjan Das. DeepSearchQA: Bridging the comprehensiveness gap for deep research agents, 2026. URL https://arxiv.org/abs/2601.20975.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel,

and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Kimi Team. Kimi K3: open frontier intelligence. *CoRR*, abs/2607.24653, 2026. doi: 10.48550/ARXIV.2607.24653. URL https://doi.org/10.48550/arXiv.2607.24653.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4745–4759, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.243. URL https://aclanthology.org/2025.naacl-long.243/.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. WebSailor: Navigating super-human reasoning for web agent, 2025. URL https://arxiv.org/abs/2507.02592.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Xinmiao Yu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Zhi-Qin John Xu, Shuai Wang, Minhao Cheng, and Jingren Zhou. WebSailor-V2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HuP16O5SJf.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

MiniMax. MiniMax M2.7: Early echoes of self-evolution, 2026. URL https://www.minimax.io/news/minimax-m27-en.

MiniMax. MiniMax M3: Frontier coding, 1m context, native multimodality. MiniMax Research blog, 2026. URL https://www.minimax.io/blog/minimax-m3.

Moonshot-AI. Kimi K2.6: Advancing open-source coding, 2026. URL https://www.kimi.com/blog/kimi-k2-6.

OpenAI. Introducing GPT-4.1 in the API, April 2025. URL https://openai.com/index/gpt-4-1/.

OpenAI. GPT-6 Astra System Card. Technical report, OpenAI, 2026. URL https://deploymentsafety.openai.com/gpt-6-astra.

OpenAI. Introducing GPT-5.5, April 2026. URL https://openai.com/index/introducing-gpt-5-5/.

OpenAI. GPT-5.6 System Card. Technical report, OpenAI, 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Pu Wu, Junkai Zhang, Baixuan Li, Haiyang SHEN, Kuan Li, Liwen Zhang, Xinyu Wang, Wentao Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebShaper: Agentically data synthesizing via information-seeking formalization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hld4TzJsnD.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85, 2014. URL http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

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

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and Ke Wang. WideSearch: Benchmarking agentic broad info-seeking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Q7YUY7zGkZ.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

Ziyi Xia, Kun Luo, Hongjin Qian, and Zheng Liu. Open data synthesis for deep research. *CoRR*, abs/2509.00375, 2025. doi: 10.48550/ARXIV.2509.00375. URL https://doi.org/10.48550/arXiv.2509.00375.

Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Z.ai. GLM-5.3. Official technical blog, 2026. URL https://z.ai/blog/glm-5.3.

Zhipu-AI. GLM-5.1: Towards long-horizon tasks, 2026. URL https://z.ai/blog/glm-5.1.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. BrowseComp-ZH: Benchmarking web browsing ability of large language models in chinese, 2025. URL https://arxiv.org/abs/2504.19314.

A VALIDATION DETAILS

Answer uniqueness. As described in Section 2.4, human review is the final stage of question filtering: annotators check factual support and question constraints, search for alternative valid answers, and revise or discard questions with identified alternatives. Nevertheless, human verification cannot exhaustively establish answer uniqueness in open-web search. Subsequent analysis of the human-authored BrowseComp benchmark likewise identified flawed question–answer ambiguous or non-unique answers (Chen et al., 2026). To account for this limitation, we asked annotators to distinguish retained questions they confidently judged to have a unique answer from those for which no alternative answer was found but uniqueness could not be fully established. The former category comprises 385 questions and forms the confirmed-unique subset in Table 4; this designation reflects evidence-based human judgment rather than a formal guarantee. We compare performance on this subset with the full benchmark to assess the sensitivity of evaluation results to residual uncertainty about answer uniqueness. Across the evaluated models, accuracy changes range from -0.41 to $+1.78$ percentage points. Rankings remain closely aligned (Kendall's $\tau = 0.9667$; Spearman's $\rho = 0.9941$), with only two adjacent model pairs changing order. These results support the stability of relative model performance when evaluation is restricted to questions with stronger human confidence in answer uniqueness.

Table 4: Accuracy (%) on the full LoHoSearch benchmark (544 questions) and the confirmed-unique subset (385 questions). Models and their order match Table 1.

Model	Full set	Confirmedunique subset
GPT-6 Astra	52.02	51.95
GPT-5.6 Sol	51.65	53.25
GPT-5.5	34.74	35.19
Gemini-3.8-Flash	48.35	50.13
Gemini-3.1-Pro	13.32	13.77
Kimi-K3	42.46	43.90
Kimi-K2.6	15.53	16.49
Claude-Opus-5	35.85	36.10
Claude-Opus-4.7	10.29	10.91
Claude-Opus-4.6	15.62	15.84
DeepSeek-V4.1-Flash	22.79	22.86
DeepSeek-V4-Flash	10.02	9.61
GLM-5.3	20.96	22.34
GLM-5.1	12.77	13.51
MiniMax-M3	2.94	3.38
MiniMax-M2.7	2.48	2.99

A.1 LLM-BASED SCORING AND VALIDATION

Dual-model scoring. We combine judgments from two complementary LLM graders: GPT-4.1 (OpenAI, 2025) with the BrowseComp grading prompt, and Qwen2.5-32B (Yang et al., 2025) with the SimpleQA grading prompt (Wei et al., 2024). The former applies stricter matching criteria that can reject correct answers containing additional qualifiers, whereas the latter accommodates semantic equivalence but can occasionally accept incorrect answers. We report the mean of the two accuracies to balance these grading tendencies.

Agreement with human review. For GPT-5.5 responses, human review yields 35.11% accuracy, close to the combined score of 34.74% and differing by only 0.37 percentage points. The individual graders yield 33.46% and 36.03%, respectively, both farther from the human result. This comparison supports averaging the two graders to obtain an aggregate score that better aligns with human assessment on the responses examined.

Two disagreement examples illustrate their complementary limitations. With reference answer "Lekki, Nigeria" and prediction "Lekki, Lagos, Nigeria", GPT-4.1 marked the response incorrect, whereas Qwen2.5-32B and human review accepted it. With reference answer "Tiga Airport" and prediction "Lord Howe Island Airport", GPT-4.1 and human review rejected the answer, whereas Qwen2.5-32B accepted it. These cases illustrate both overly strict rejection of a correct answer and erroneous acceptance of a different entity.

Judge sensitivity. We examine how the choice of grader affects performance rankings across the following evaluated models: GPT-5.5, DeepSeek-V4-Pro and V4-Flash, Claude-Opus-4.6 and 4.7, Kimi-K2.6, Gemini-3.1-Pro, GLM-5.1, LongCat-Flash-Thinking-2601, and MiniMax-M2.5 and M2.7. Relative to rankings based on the mean accuracy of both graders, GPT-4.1 with the BrowseComp prompt yields Kendall's $\tau = 0.8545$ and Spearman's $\rho = 0.9545$, while Qwen2.5-32B with the SimpleQA prompt yields $\tau = 0.9297$ and $\rho = 0.9818$. These correlations indicate that relative model performance remains broadly stable across the two graders.

B Calibration Error

We adopt Expected Calibration Error (ECE) to measure the alignment between model confidence and actual accuracy. Predicted probabilities are partitioned into five equally spaced bins: $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1.0]$. ECE is defined as:

$$\text{ECE} = \sum_{i=1}^{K} \frac{n_i}{N} \left|\text{acc}(i) - \text{conf}(i)\right| \tag{2}$$

where K is the number of bins, N is the total number of samples, n_i is the number of samples in the i-th bin, $\text{acc}(i)$ is the empirical accuracy of the i-th bin, and $\text{conf}(i)$ is the average predicted confidence in the i-th bin. A lower ECE indicates better calibration between model confidence and actual performance.

C System Prompt

We adopt the same system prompt as BrowseComp to instruct all models. The prompt content is as follows:

```
Your response should be in the following format:
Explanation: {your explanation for your final answer}
Exact Answer: {your succinct, final answer}
Confidence: {your confidence score between 0% and 100% for your answer}
```

D Tool Definitions

All models are equipped with two tools: `search` and `browse`. Their definitions are as follows:

```
{"type": "function",
 "function": {
   "name": "search",
   "description": "Web search, using traditional
     search engines like google, complex
     questions need to be broken down into
     simple queries",
   "parameters": {
     "type": "object",
     "properties": {
       "query": {
         "type": "array",
         "items": {"type": "string"},
         "description": "List of search queries.
           You can search up to 5 queries at
           the same time."}
       },
     "required": ["query"]
   }
 }
}
```

```
{"type": "function",
 "function": {
   "name": "browse",
   "description": "Retrieve detailed content
     from one or more specified web pages by
     providing their URLs.",
   "parameters": {
     "type": "object",
     "required": ["url"],
     "properties": {
       "url": {
         "type": "array",
         "description": "List of urls. You can
           browse up to 3 webpages at the
           same time.",
         "items": {"type": "string"}
       }
     }
   }
 }
}
```

E Case Studies

We present one tree-structured and one graph-structured example from LoHoSearch.

Case 1 (Tree-structured): Consider the following facts. They all pertain to one album. Which album is it?
1. It includes singles from another album (Album A), which was created by a group formed on a reality competition television series that originally aired from the early 2000s to the mid-2010s; additionally, Album A includes a solo song by a person (Person D) who has one child, works as a pop singer and television host, and was born in the late 1970s.
2. It includes a track written by a person (Person B) who shares the same name as a person (Person E) who earned an award in the early 2000s; this award has been presented annually since the year 2000 and has an alternative name that includes an acronym for a community service initiative.
3. It includes tracks produced by a person (Person C) who studied an academic discipline historically defined by roughly three main methods—individual casework, social group work, and community intervention work—and from the late 20th century, many different methods derived from these three classic methods.
Answer: The Best of No Angels

Case 2 (Graph-structured): Who is described by the following conditions?
1. This person toured with a musical group (Group C). Group C performs in a music genre (Genre D) that is featured at a music festival (Festival E), where Group C was a performer in the early 2010s. The group also released an album (Album H) in the early 2010s, which includes a musical composition (Composition G) by them. Additionally, Group C performs at a recurring event (Event F), and this person has performed at Event F as well.
2. They recorded a comedy album at a recurring event (Event I). Event I featured Group C as a guest of honor in the late 2010s.
3. Furthermore, this person appeared as a guest on a podcast (Podcast B) and wrote and performed shows at a film festival (Festival J).
Answer: Joseph Scrimshaw

F Construction Settings

Table 5 lists the sampling parameter values used in Section 2.2. For tree-structured subgraphs, each root connects to three intermediate entities, and each intermediate entity connects to at most two leaf nodes. The candidate-set criterion $|\mathcal{S}| > \tau$ uses $\tau = 3$, requiring at least four candidates per selected relation. Graph-structured subgraphs contain at most ten entities, at least three distinct entity types, and at least eleven edges. We further require at least two confounding candidates other than the answer that share its type and connect to entities of every type represented among its neighbors.

Table 5: Parameter settings for subgraph sampling.

Parameter	Meaning	Value
N	Intermediate entities per root (tree)	3
M	Maximum leaf nodes per intermediate entity (tree)	2
τ	Candidate-set size threshold (tree)	3
K	Maximum number of entities (graph)	10
$D_{\min}$	Minimum number of distinct entity types (graph)	3
$L_{\min}$	Minimum number of edges (graph)	11
$C_{\min}$	Minimum number of confounding candidates (graph)	2