AgentSearchBench:
Evaluating Agentic Search with Agent-as-a-Judge

Anonymous Authors[1]

Abstract

Agentic search such as Deep Research systems—where large language models autonomously browse the web, synthesize information, and return citation-backed answers—represents a major shift in how users interact with web-scale information. While promising greater efficiency and cognitive offloading, the growing complexity and open-endedness of agentic search have outpaced existing evaluation benchmarks and methodologies, which largely assume short horizons and static answers. In this paper, we introduce AgentSearchBench, a benchmark of 100 realistic, high-quality, long-horizon tasks that require real-time web interaction and extensive information synthesis. To address the challenge of evaluating time-varying, multi-source answers, we propose a novel Agent-as-a-Judge framework. Our method leverages task-specific, tree-structured rubrics and rubric-based judge agents to automatically assess both factual correctness and source attribution with a high agreement with humans. We conduct a comprehensive evaluation of 9 frontier agentic search systems and human performance, and a detailed error analysis to draw insights for future development. Together, AgentSearchBench and our evaluation framework provide a rigorous foundation for developing and benchmarking the next generation of trustworthy, high-capability agentic search systems.

1. Introduction

Web search has long been the gateway to the world's knowledge, underpinning everything from everyday fact-checking to frontier scientific discovery. The core techniques supporting web search have undergone constant evolution in the past decades, from TF-IDF (Salton et al., 1975) for term statistics to PageRank (Brin & Page, 1998) for network analysis and learning to rank (Liu et al., 2009; Burges et al., 2005) for supervised learning. Yet the core interaction model has remained essentially unchanged: users issue a query, receive a ranked list of URLs, and must manually open, read, and synthesize multiple pages to answer complex questions. Current web search is inherently *user-driven*: it retrieves pieces of information but relies on users to interpret and assemble those pieces. That places a significant cognitive load on users, especially as the complexity of the digital world grows.

Recent advances in large language models (LLMs) have sparked the development of *agentic search* systems. Rather than taking keyword queries and returning a list of links, an agentic search system can decompose and plan for complex queries, iteratively search the web and interact with dynamic websites, and synthesize information into a citation-backed response. In recent years, agentic search has quickly progressed from *search-augmented LLMs* (e.g., ChatGPT/Perplexity Search) to LLM-based *autonomous web agents* (Nakano et al., 2021; Deng et al., 2023; Zhou et al., 2024; Zheng et al., 2024; Anthropic, 2024; OpenAI, 2025b) and recently *Deep Research* systems (Google, 2025; OpenAI, 2025a) specifically optimized for long-horizon browsing and search behavior. By off-loading many low-level tasks, such as query decomposition and reformulation, web browsing, and basic analytics, to a tireless AI agent, agentic search promises to empower human users to focus their cognitive capacity on more important matters like oversight and critical decisions, improving both search efficiency and quality.

However, the rapidly growing complexity of agentic search systems and their tasks is leading to an *evaluation crisis*: how to evaluate the result of a long-horizon task that an AI agent or human produces after taking possibly half an hour and hundreds of actions across dozens of websites? Meanwhile, automatic and reliable evaluation has proven crucial for the iterative development of AI technologies, especially in the early stages (Hendrycks et al., 2021; Chiang et al., 2024; Yue et al., 2024). For agentic search, evaluation is

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

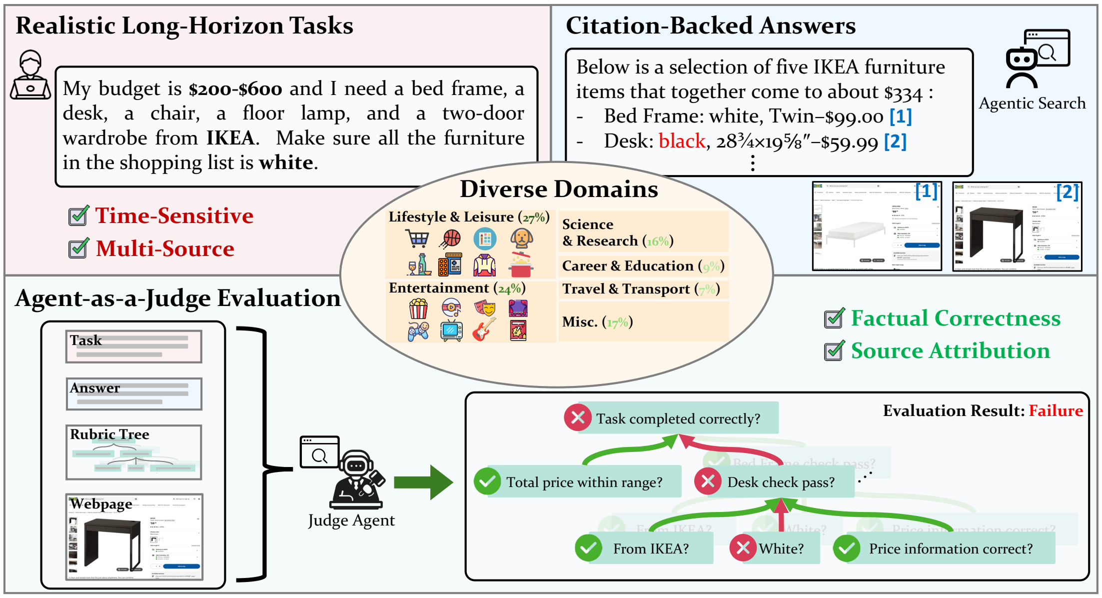


Figure 1: AgentSearchBench features realistic and diverse long-horizon tasks and a novel Agent-as-a-Judge framework based on rubric trees to evaluate complex, time-varying, and citation-based answers.

also critical for establishing its *trustworthiness*—while traditional search requires the user to read original documents and verify information, an agent that synthesizes answers must be relied on to be correct and unbiased. Automatic evaluation serves as the first line of defense to detect whether an agent is just hallucinating plausible-sounding answers or the cited sources verifiably back them.

Existing benchmarks and evaluation methodologies struggle to keep up with the growing complexity of agentic search. Many benchmarks have been proposed for autonomous web agents (Deng et al., 2023; Yao et al., 2022; Zhou et al., 2024; Lu et al., 2024; Xue et al., 2025) but they primarily focus on tasks of a moderate horizon (e.g., around 10 actions) that can be completed on a single website. Several benchmarks cover cross-website search tasks (Mialon et al., 2023; Yoran et al., 2024; Song et al., 2025), including most recently BrowseComp (Wei et al., 2025) from OpenAI. However, to facilitate automatic evaluation, a common compromise was made: they focus on tasks with *predefined, time-invariant answers*, oftentimes just a single answer string. While these benchmarks still provide valuable signals for evaluating certain aspects of agentic search systems, they are far from the full spectrum of tasks that current and future systems are faced with. Consider an everyday task already within reach of current Deep Research systems, shown in Figure 1. It does not have a predefined answer but requires interacting with live websites to get real-time information. A corresponding agent trajectory may span dozens to hundreds of actions on the IKEA website, let alone more complex tasks that span many websites. We need new evaluation methodologies and benchmarks for such *long-horizon, time-varying* tasks.

In response to these challenges, we propose AgentSearchBench, a new benchmark designed to rigorously evaluate agentic search systems on realistic and long-horizon tasks involving real-time web search and browsing. It consists of 100 high-quality tasks across diverse practical domains. Each task has undergone multiple stages and hours of expert labor for polishing and validation to ensure its realism, complexity, and verifiability. Agentic search systems typically produce long, time-varying answers (e.g., the product catalog of a shopping website constantly changes) ranging from hundreds to thousands of words on these tasks. The complexity is far beyond what conventional LLM-as-a-Judge (Zheng et al., 2023) methods are used for.

We propose a novel *Agent-as-a-Judge* framework to automatically yet reliably evaluate such complex answers. The key insight behind our evaluation methodology lies in the *generation-verification asymmetry*: while the generated answers can vary substantially across agents, search strategies, or query times, we know *a priori* what each task is looking for and can design a *task-specific rubric* to specify the evaluation logic. We propose a tree-structured rubric and a human-in-the-loop pipeline for rubric generation. At a high level, a rubric evaluates two main aspects of an answer: *correctness* (i.e., whether the answer satisfies all the requirements of the task) and *attribution* (i.e., whether each fact in the answer can be attributed to the cited source). At the operational level, a rubric tree breaks down the evaluation into hierarchical evaluation nodes, where each leaf node conforms to a binary judgment and the internal nodes aggregate and propagate the results toward the root following various aggregation logic. Given a rubric tree, we develop a *task-specific judge agent*, an agentic workflow interleaving LLM-based information extraction, LLM-as-a-Judge, and tool calls following the rubric, to automatically evaluate complex answers from agentic search systems (see Figure 1 for illustration).

We conduct evaluations on nine frontier agentic search systems and perform a human study for comparison. Our results show a substantial performance gap between humans and current agentic search systems: even the most advanced system achieves only a 21% success rate. Current Deep Research systems, despite their strengths in generating comprehensive responses and performing tasks over extended periods, frequently produce incomplete results, fail to retrieve necessary information, or hallucinate synthesized answers. Nonetheless, these systems demonstrate initial promises in tackling realistic, long-horizon search tasks, highlighting the potential and future directions for agentic search.

2. Related Work

Agentic Search. We define *agentic search* as systems that iteratively and autonomously tackle complex search tasks using a combination of tools (e.g., search APIs, retrievers, or web browsing). The autonomy is typically powered by an LLM that decomposes the initial search task, dynamically reasons and plans based on new information, or interacts with live websites. Early systems like MindSearch (Chen et al., 2024b), ChatGPT and Perplexity Search augment LLMs with search APIs to iteratively search for up-to-date information. However, solely relying on conventional web search also inherits its limitations. For example, many websites dynamically render information not indexed by search engines based on user interaction. Autonomous web agents (Nakano et al., 2021; Deng et al., 2023; Yao et al., 2022; Zhou et al., 2024), especially those with visual perception of the web (Zheng et al., 2024; Koh et al., 2024; Gou et al., 2025; Qin et al., 2025), have emerged to browse the real-time web as humans do. OpenAI's Operator (OpenAI, 2025b), with specialized reinforcement learning training, represents the current frontier (Xue et al., 2025). Recent advances in reasoning models (Jaech et al., 2024; Guo et al., 2025) have enabled the development of Deep Research systems (OpenAI, 2025a; Google, 2025; Hugging Face, 2025) that leverage a suite of advanced tools, including search APIs and web browsing, to conduct substantially longer-horizon and deeper research on complex topics. However, there is yet a benchmark designed to simultaneously evaluate this broad spectrum of agentic search systems, a gap that our work aims to bridge.

Benchmarks and Evaluation Methodologies. Most existing benchmarks for web agents focus on evaluating whether an agent can autonomously perform certain processes on a single website (Deng et al., 2023; Yao et al., 2022; Zhou et al., 2024; Lu et al., 2024; He et al., 2024; Xue et al., 2025; Koh et al., 2024). The tasks tend to be short (e.g., less than 10 actions) and transactional (e.g., purchasing a flight ticket). Therefore, they can be useful for evaluating the web browsing aspect of agentic search but not the whole systems. Several recent benchmarks have a stronger focus on search over the open web (Mialon et al., 2023; Yoran et al., 2024; Wu et al., 2025; Song et al., 2025; Wei et al., 2025). However, for the feasibility of automated evaluation, these benchmarks have made a common compromise: they limit the benchmark to tasks with *predefined, time-invariant answers*, oftentimes just a single answer string. The BrowseComp benchmark (Wei et al., 2025) from OpenAI, a concurrent work to ours, is representative of this evaluation methodology. Similar to ours, it also leverages the generation-verification asymmetry. It specifically targets tasks that are *hard to solve but easy to verify* (e.g., the answer is a unique unambiguous string but may require combing through hundreds of webpages to find it). This strategy is adopted to sidestep the challenge of automatically evaluating complex, time-varying answers, but at the cost of systematically deviating from the true user query distribution. In contrast, we take this challenge head-on with a novel Agent-as-a-Judge methodology. That allows our benchmark to include more realistic and complex tasks that expect a comprehensive answer with real-time information.

LLM-as-a-Judge (Zheng et al., 2023) has been widely used in evaluating complex tasks, including for web agents (Pan et al., 2024a; He et al., 2024; Xue et al., 2025). However, the complexity of agentic search is far beyond what a few LLM calls can evaluate, necessitating an Agent-as-a-Judge approach (Zhuge et al., 2024; Starace et al., 2025). PaperBench (Starace et al., 2025) (a concurrent work) is most related to ours in that it also adopts a tree-structured rubric, though it is manually written by human experts and used to evaluate replication of AI research. Our work goes further by largely automating the generation of rubrics. We also have more sophisticated score aggregation methods beyond simple weighted averaging due to the diversity of our tasks. Finally, our attribution evaluation is also related to the attribution literature (Yue et al., 2023; Gao et al., 2023; Li et al., 2024; Liu et al., 2023).

3. AgentSearchBench

3.1. Overview

We introduce AgentSearchBench, a novel benchmark designed to rigorously evaluate agentic search systems on realistic and complex information-gathering tasks involving real-time web search and browsing. There are two main challenges in constructing such a benchmark:

- *How to collect sufficiently complex yet realistic tasks?*

- *How to automatically and reliably evaluate the complex answers generated by different agentic search systems?*

In §3.2, we discuss a three-stage process we adopt to propose, refine, and validate tasks. We spend hours of expert labor on each task to ensure its validity, diversity, clarity, and verifiability. To tackle the significant evaluation challenge, we propose a novel Agent-as-a-Judge framework that evaluates both the *correctness* (i.e., whether the answer satisfies all the requirements of the task) and *attribution* (i.e., whether each claim in the answer can be attributed to the cited source) of an answer. Specifically, we describe the design of our rubric tree in §3.3 and the rubric-based judge agent in §3.4. Finally, we show benchmark statistics in §3.5.

3.2. Task Collection

Tasks in AgentSearchBench shall have the following characteristics: (1) *Realistic and diverse*. Tasks must reflect practical user needs in diverse domains, providing substantial real-world value when solved; (2) *Long-horizon and laborious*. Tasks require substantial human effort due to an extended length and breadth of the required searches; (3) *Objective and verifiable*. Each task must have clearly defined evaluation criteria that are verifiable by checking the corresponding provenance (i.e., the cited webpages). (4) *Time-sensitive.* The answer to a task can change over time, although it is not a requirement for every task.

Our task collection team consists of three groups of annotators (all are experienced computer science students or professionals): *task proposers*, *refinement experts*, and *validation experts*, who lead different stages of the procedure. First, *task proposers* freely generate task ideas based on their authentic search needs or inspirations from our provided domain guidelines, ensuring initial alignment with the realism and laboriousness desiderata. Second, trained *refinement experts* iteratively revise or filter tasks to enforce strict verifiability while collaborating closely with the original task proposers to maintain task relevance. Third, experienced *validation experts* manually attempt and verify each refined task, ensuring feasibility, determinism, and clarity of all the evaluation criteria. Only tasks independently validated by at least two validation experts are included in AgentSearchBench.

3.3. Rubric Tree

Our rubric-tree structure includes two main types of nodes. Each node is classified either as a *critical node*, representing essential criteria whose failure immediately fails its parent node (e.g., the budget evaluation node (a) in Figure 2), or as a *non-critical node*, allowing partial scoring (e.g., we independently assess each of the five requested furniture items and give partial credits in Figure 2). Additionally, a small subset of nodes are marked as *sequential*, reflecting logical dependencies where a failure in an earlier step *short-circuits* subsequent evaluations. For example, if a task requires finding a certain paper and subsequently the email of the first author, failing to find the correct paper makes it pointless to evaluate the subsequent email node.[1] Each leaf node represents a criterion for answer correctness that can be assessed through straightforward verification, yielding a binary score of 0 or 1. These binary scores are then aggregated by parent nodes to determine the scores for higher-level criteria.

Formally, let v be a node in the rubric tree and $C(v)$ its child nodes. We partition child nodes into critical nodes $K(v) \subseteq C(v)$, and non-critical nodes $N(v) = C(v) \setminus K(v)$. The score $s(v) \in [0, 1]$ of v is recursively defined as:

$$s(v) = \begin{cases} 0, & \text{if } \exists\, u \in K(v) \text{ s.t. } s(u) = 0, \\ \dfrac{1}{|N(v)|} \displaystyle\sum_{u \in N(v)} s(u), & \text{if } \forall\, u \in K(v)\ s(u) = 1 \text{ and } |N(v)| > 0, \\ 1, & \text{otherwise.} \end{cases}$$

Intuitively, the score aggregation employs a *gate-then-average* strategy: critical nodes serve as gating conditions when paired with non-critical nodes. In practice, critical nodes often represent basic and essential constraints rather than incremental progress, thus their scores do not directly contribute to the averaging process for partial scoring, but instead function solely to validate the meaningfulness of aggregating scores from non-critical nodes. Finally, if a node only contains critical child nodes, which indicates that each child represents a necessary condition for the parent criterion, the score of the parent node directly depends on the passing of all these critical child nodes (e.g., in Figure 2, the wardrobe node b gets a score 1 only if all the child nodes pass; otherwise 0.)

We define two metrics based on the final aggregated score at the root node: 1) **Success Rate**, the percentage of tasks where the root node gets a perfect score of 1, which means the task was fully completed and all the criteria are met, and 2) **Partial Completion**, the average of the root node scores over all the tasks, which corresponds to the percentage of passed fine-grained evaluation nodes.

3.4. Rubric-based Judge Agent

Following the rubric tree defined in §3.3, each task in AgentSearchBench is evaluated by a dedicated *judge agent*, implemented as a Python script with an agentic workflow following the task-specific rubric. Each judge agent takes the submitted answer text (including the provenance) as input, first evaluates each leaf node, and then aggregates and propagates scores upwards to produce a final score at the root of the rubric tree.

[1]The sequential logic is sufficient for our current tasks, though future work can explore other logic.

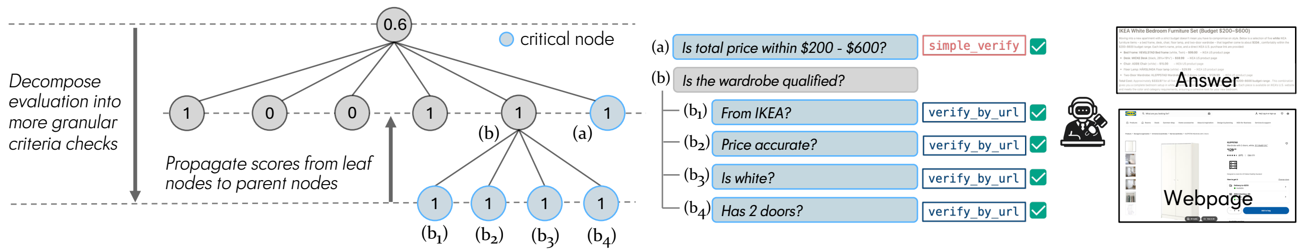


Figure 2: Rubric-tree evaluation with *agent-as-a-judge*. (Left) Top-down, task goals are decomposed into a tree; bottom-up, binary scores from leaf nodes propagate to form the overall task score. (Right) The judge scores each node based on its criterion— `simple_verify` if the answer alone suffices, or `verify_by_url` if the answer is backed by citation. See more discussions in §3.3 and §3.4.

A judge agent uses two LLM-based tools for leaf node evaluation: (1) an *Extractor* that parses the raw answer to extract structured information (e.g., item names, prices, and URLs), and (2) a *Verifier* that judges factual correctness and attribution. Take the leaf node b_3 in Figure 2 as an example, the Extractor extracts the corresponding bit of information from the raw answer (i.e., the identified wardrobe is claimed to be white), and the Verifier examines the screenshot of the corresponding webpage to determine if it is indeed true. We use OpenAI `o4-mini` for both tools.

Manually crafting such judge agent scripts from scratch is prohibitively demanding due to the complexity and granularity of the evaluation criteria. Thus, we first develop a modular Python toolkit encapsulating reusable rubric-management utilities and standardized *Extractor* and *Verifier* modules. This toolkit substantially reduces coding overhead, allowing annotators to focus primarily on rubric design rather than code details. Nonetheless, script creation remains challenging even with this toolkit. To further facilitate the development, we build an automated LLM-based script generation pipeline that produces an initial version of the script based on the task description. The generated scripts undergo iterative autonomous refinements (including self-debug (Chen et al., 2024a) and self-reflection (Shinn et al., 2023; Madaan et al., 2023)) to auto-correct minor or common errors. Finally, scripts are rigorously validated through a two-stage human refinement process, improving correctness and generalizability to all possible answers. Further details about rubric and script development is provided in Appendix C.

3.5. Benchmark Statistics

Through the pipeline described §3.2-§3.4, we collect a total of 100 carefully curated tasks, each accompanied by a carefully validated rubric tree and a judge agent. After initial proposals, each task undergoes on average more than two hours of additional human validation. Task distribution across domains is shown in Figure 1.

Table 1: Benchmark statistics.

(a) Rubric complexity w.r.t. the number of nodes and tree depth.

	Avg	Min	Max
# Leaf nodes	35	3	189
# Total nodes	54	4	267
Depth	4	2	5

(b) Human effort required per task (Subset-25).

	Avg	Min	Max
Time (min)	17	3	55
# Websites	6	3	24
# Webpages	104	21	448

With our automated script-generation pipeline, most judge agents only require minimal human revision (zero to two rounds instruction-guided re-generation), although particularly complex rubrics (e.g., those with over 50 evaluation nodes) still often demand about half to one hour of additional manual revision. The statistics of the rubric trees in Table 1 (a) show the complexity of our tasks, with rubric trees having up to 5 layers and 267 evaluation nodes.

To further quantify the complexity of our benchmark, we conduct a human study on a representative subset of 25 tasks (Subset-25). Seven annotators are asked to manually complete these tasks (each task by three different annotators), allowing us to observe human behaviors and measure the human effort associated with our tasks. Results in Table 1 (b) show that our tasks are indeed highly time-consuming for humans: it can take up to one hour (an underestimate since we set a time limit of one hour) and visit a whopping 24 websites and 448 webpages to get the answer.

Table 2 shows the comparison of AgentSearchBench to other related benchmarks. Echoing the discussion in §2, AgentSearchBench is the only agentic search benchmark to date focusing on long-horizon, time-sensitive tasks, thanks to our advanced Agent-as-a-Judge evaluation. It is worth

Table 2: Comparison with existing benchmarks for web browsing or search on live websites. **Horizon**: average number of required actions per task. Short (<10), Medium ($10-50$), Long (>50). **Time-Sensitive**: whether the answer will change over time.

	Horizon	# of Tasks	Time-Sensitive	Evaluation
Online-Mind2Web (Xue et al., 2025)	Short	300	✓	LLM-as-a-Judge
WebVoyager (He et al., 2024)	Short	643	✓	LLM-as-a-Judge
Mind2Web-Live (Pan et al., 2024b)	Short	542	✓	Rule
BEARCUBS (Song et al., 2025)	Short	111	✗	Manual Evaluation
WebWalkerQA (Yoran et al., 2024)	Short	680	✗	Answer Match
GAIA (Mialon et al., 2023)	Medium	466	✗	Answer Match
AssistantBench (Yoran et al., 2024)	Medium	214	✗	Answer Match
BrowseComp (Wei et al., 2025)	Long	1,266	✗	Answer Match
AgentSearchBench	Long	100	✓	Agent-as-a-Judge

noting that, even though there are only 100 tasks, each task contains dozens to hundreds of fine-grained evaluation nodes, thus still providing sufficient differentiation power.

To reduce the risk of data contamination and our judge agents being abused as reward models for reinforcement learning, we plan to split our benchmark into a *public development set* (10 tasks), that include both the task descriptions and evaluation scripts, and a *private test set* (90 tasks), that only include the task descriptions. We will maintain a leaderboard and participants are required to submit their answers to the leaderboard to be evaluated by us. We have sufficient initial funding and will seek additional sponsorship to sustain the leaderboard.

4. Experiments

4.1. Experimental Setup

We evaluate leading agentic search systems on AgentSearchBench. Given the complexity of our tasks, our primary evaluation metric is **Partial Completion**, defined in §3.3. We also report the **Success Rate**, which only considers answers achieving full completion as successful.

We independently run each system three times on each task and report the average Partial Completion, Success Rate, and their standard deviations. We also report **Pass@3**, i.e., a success is counted if any of the three attempts at a task succeeds. Moreover, we record the average completion time and the average answer length as two behavioral factors reflecting the user experience of the systems.

4.2. Baselines

We evaluate a broad set of cutting-edge agentic search systems. We choose ChatGPT Search (OpenAI, 2024) and Perplexity Pro Search (Perplexity AI, 2024) as two high-profile common search product, and a group of Deep Research systems (Hugging Face, 2025; Perplexity AI, 2024; xAI, 2025; Google, 2025; OpenAI, 2025a) that are optimized for long-horizon search, some could even work for more than half an hour for a user query. Finally, we evaluate OpenAI Operator (OpenAI, 2025b) as the frontier autonomous web agent that browses and searches by directly operating on graphical user interfaces. Because of the complexity of the tasks in our benchmark, we only include frontier systems that are sufficiently capable and can reliably provide answer attribution in the responses.

To gain further insights into the practical value of these systems, we also conduct a human study for comparison, as previously described in §3.5, where we ask human annotators to manually attempt the tasks under fair settings (see details in Appendix D).

4.3. Main Results

As shown in Table 3, the realistic and long-horizon nature of AgentSearchBench's tasks poses huge challenges for these systems. All evaluated systems achieve low Partial Completion scores and Success Rates, highlighting substantial room for improvement in current agent capabilities.

Human annotators achieve the highest performance. However, although these tasks are straightforward for humans given sufficient time and patience, annotators often miss or mistakenly omit critical details in practice. This frequently results in a partial score around 60-90%, which, on the other hand, underscores a valuable opportunity for agentic systems to assist with. Notably, humans also take the longest time to obtain the answers.

Current agentic systems still fall short in accurately completing such long-horizon tasks. *Web agent systems* (e.g., Operator in this work), which interact with web pages in a human-like manner, show effectiveness on short and clearly defined web interactions. However, for tasks requiring hundreds or even thousands of browser actions, Operator quickly accumulates errors in earlier stages and fails to recover. Moreover, unlike Deep Research systems specifically

Table 3: Main evaluation results. We report the partial completion score, full-task success rate, pass@3, average time (in minutes), average answer length (in words), and their standard deviation. *: To reduce human workload, the human study is conducted on the Subset-25 as described in Table 1.

	Partial Completion	Success Rate	Pass@3	Time (min)	Answer Length
ChatGPT Search	$0.20_{\pm 0.01}$	$0.05_{\pm 0.02}$	0.04	< 1	$303_{\pm 116}$
Perplexity Pro Search	$0.25_{\pm 0.05}$	$0.11_{\pm 0.02}$	0.12	< 1	$390_{\pm 150}$
OpenAI Operator	$0.14_{\pm 0.01}$	$0.01_{\pm 0.01}$	0.02	$9.62_{\pm 4.47}$	$124_{\pm 85}$
HF Open Deep Research	$0.19_{\pm 0.10}$	$0.04_{\pm 0.04}$	0.08	$13.82_{\pm 7.14}$	$201_{\pm 159}$
Perplexity Deep Research	$0.32_{\pm 0.05}$	$0.08_{\pm 0.04}$	0.19	$5.83_{\pm 1.78}$	$501_{\pm 227}$
Grok DeepSearch	$0.31_{\pm 0.08}$	$0.11_{\pm 0.05}$	0.16	$2.43_{\pm 0.94}$	$1{,}406_{\pm 357}$
Grok DeeperSearch	$0.34_{\pm 0.03}$	$0.14_{\pm 0.03}$	0.20	$5.56_{\pm 1.59}$	$1{,}347_{\pm 347}$
Gemini Deep Research	$\underline{0.37}_{\pm 0.03}$	$\underline{0.15}_{\pm 0.02}$	$\underline{0.24}$	$7.49_{\pm 2.23}$	$2{,}816_{\pm 1{,}122}$
OpenAI Deep Research	$\mathbf{0.44}_{\pm 0.06}$	$\mathbf{0.21}_{\pm 0.08}$	**0.26**	$9.31_{\pm 2.75}$	$540_{\pm 446}$
Human*	$0.68_{\pm 0.20}$	$0.33_{\pm 0.17}$	0.64	$17.16_{\pm 8.32}$	$193_{\pm 203}$

optimized for comprehensive search and response synthesis, Operator often neglects to provide sufficient provenance for facts even when explicitly prompted to do so. Thus, despite their human-like web interaction capabilities—which are indeed essential for some of our tasks—web agents still lack the long-term reasoning, planning, and memory required to reliably complete the tasks in our benchmark.

In contrast, *Deep Research* systems, which are trained or prompted explicitly for comprehensive question-answering and often also equipped with web interaction abilities, demonstrate stronger performance. However, despite substantial increases in inference time and generated response length compared to quicker solutions such as ChatGPT Search or Perplexity Pro, performance gains remain moderate. For instance, Grok DeeperSearch, while utilizing more than double the inference time of Grok DeepSearch, only achieves approximately a 0.03 improvement in Partial Completion and Success Rate. Additionally, over-training or excessive inference time allocation seem to produce problematic results: Gemini Deep Research frequently generates lengthy and overly detailed document-style answers, even for straightforward tasks, without clear performance advantages. These observations suggest merely increasing inference time and output length does not fundamentally resolve underlying capability limitations. The open-source solution, HF Open Deep Research, although achieving high performance on benchmarks like GAIA (Mialon et al., 2023), still fail to complete the more complex tasks in AgentSearchBench efficiently and accurately. Based on off-the-shelf models, it substantially lags behind other Deep Research systems, suggesting the importance of optimizing the underlying LLM for enhancing long-horizon search capabilities.

Among the evaluated systems, OpenAI Deep Research stands out as the best-performing agent across nearly all the dimensions. It achieves high Partial Completion scores without excessive inference time or overly verbose responses, suggesting a more balanced and effective system for long-horizon tasks.

4.4. Error Analysis

We conduct a detailed error analysis to gain deeper insights about the failure modes. We ask human annotators to manually find identifiable errors in in the answers from Subset-25 , independent from the results from judge agents to do an unbiased examination. Considering the common failure patterns as well as saving human workload, we pre-define and capture the following error types, primarily under two dimensions: *answer text check* and *attribution check*:

Answer Text Check. We evaluate the textual completeness and correctness of the answers, independently of attribution verification. It includes the following subcategories:

- **Incompleteness**: The answer explicitly fails to fully satisfy the task, with two subcategories: (1) *Information Not Found*: The agent explicitly states an inability to find the requested information. (2) *Partial Missing*: The answer contains fewer items or steps than explicitly requested by the task.
- **Criteria Violation**: The answer explicitly contradicts the clearly stated task criteria or provides incorrect factual information, identifiable directly from the answer text itself (excluding incompleteness). Examples include providing an item priced higher than a stipulated threshold or incorrectly identifying a specified follow-up work.

Attribution Check. Independent of textual completeness, we verify whether the provided URL sources support the key information stated in the answer. These are often related to hallucinations from the LLM-based agent systems:

- **Invalid Attribution**: URLs provided by the agent are expired, malformed, or fabricated.
- **Missing Attribution**: No URL is provided for supporting

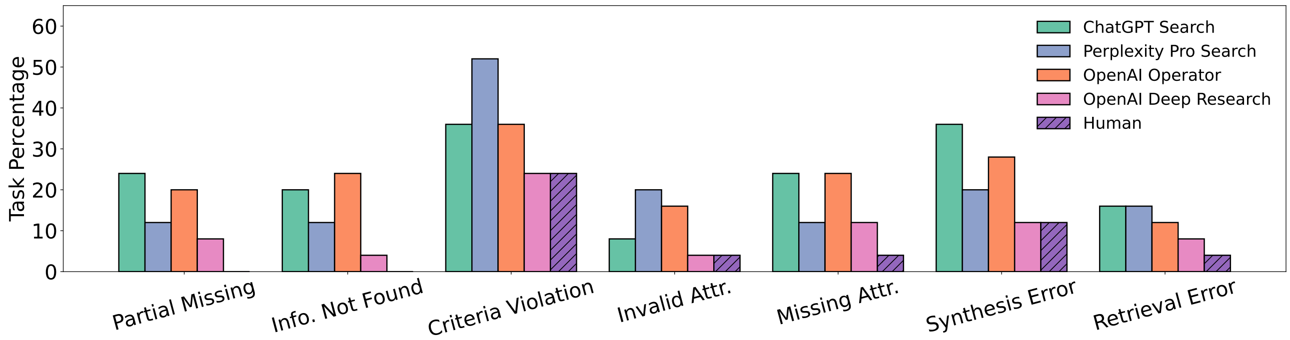


Figure 3: Error breakdown across agents and humans. Bars indicate the percentage of tasks exhibiting each error type. We compare answers from four automated systems and human completion (hatched). Lower bars indicate fewer mistakes.

the claims made.

• **Unsupported Answer**: URLs do not substantiate the claims. This error further splits into: (1) *Synthesis Error*: The URL correctly contains the required information, but the agent misrepresents or incorrectly extracts this information in the generated text. (2) *Retrieval Error*: The provided URLs are irrelevant or incorrect and thus fail to match the claims made in the answer.

As shown in Figure 3, among all evaluated systems, including human annotators, the most prevalent error category identified is *Criteria Violation*. Even human annotators provide answers explicitly contradicting certain criteria in 24% of tasks. This is largely due to the tedious nature of long-horizon search tasks, where humans, when dealing with hundreds of pages and criteria, also cannot make sure perfect accuracy and completeness. Agentic search systems have great potential to assist humans and offload the cognitive workload.

Besides, hallucinations occur consistently in both answer texts and attributions by current agentic search systems, with the most affected system showing at least one type of hallucination in 56% of tasks. These systems frequently distort accurate information from valid sources (*synthesis error*), provide claims without necessary attribution (*missing attribution*), and sometimes even fabricate attribution sources (*invalid attribution*). Another small proportion of errors arises from the retrieval capabilities of agentic search systems, manifested as *retrieval errors*.

Notably, a significant discrepancy between humans and agentic systems emerges regarding incompleteness. Agentic search systems often exhibit laziness, failing to fully complete tasks independent of answer correctness—a rare issue among human users. This issue is particularly pronounced among common search products and web agents, likely due to limitations in sustained instruction-following abilities, especially during long-horizon tasks involving numerous sequential steps and extensive web browsing. On the contrary, deep research systems, designed to conduct long-horizon research and provide comprehensive reports, suffer less from this issue.

4.5. Human Agreement

To validate our rubric's generalization, we conduct a human agreement study across 10 tasks. For each task, we randomly select two agent answers that are held out during rubric development. Annotators unfamiliar with the rubric are asked to: (1) assess agreement with the rubric tree, and (2) manually verify leaf-node evaluations if they agree. Human annotations are compared to judge agent results. Annotators generally agree with the rubrics; however, two annotators suggest slight modifications to two rubrics for stricter evaluations. At the node level, we observe 4 discrepancies out of 201 nodes. Two are due to human oversight, one is a judge agent error in identifying a webpage detail, and one is due to overly strict judgment by the agent. These results indicate our rubric-based evaluations align reasonably well with human judgments, as a reliable and scalable evaluation method.

5. Conclusions

In this work, we introduced AgentSearchBench, a novel benchmark specifically designed for comprehensively evaluating agentic search systems on long-horizon information-gathering tasks. We proposed a flexible, reliable, automated, and scalable evaluation framework based on Agents-as-a-Judge that systematically assesses agent performance on open-ended long-horizon search tasks. Our comprehensive empirical analysis spanning AI-based search engines, deep research systems, and web agents reveals significant gaps between current state-of-the-art systems and human-level performance. AgentSearchBench serves as a valuable resource and rigorous assessment platform for better advancing agentic search systems.

Impact Statement

In this section, we discuss broader impacts from two interconnected perspectives: the broader implications of agentic search systems, and the impacts associated with the release and use of the AgentSearchBench benchmark.

Agentic Search Systems. Advanced agentic search systems promise a transformation in how users interact with the web, shifting from manual, multi-step information gathering to streamlined, automated information synthesis. This change could significantly reduce cognitive load, improve efficiency, democratize sophisticated search capabilities, and support informed decision-making across diverse fields including education, healthcare, commerce, and policy-making.

Despite benefits, enhanced agentic search may exacerbate misinformation by generating seemingly credible yet incorrect or unsupported information. Malicious actors could exploit such systems for large-scale disinformation or unauthorized data extraction. Additionally, agentic systems risk perpetuating existing biases found in web content, raising fairness concerns and potentially leading to discriminatory outcomes without careful oversight and transparency.

AgentSearchBench Benchmark. By emphasizing rigorous evaluation through structured rubrics and explicit verification of source fidelity, AgentSearchBench facilitates the development of transparent and accountable agentic search systems. Establishing standardized, robust evaluation practices helps accelerate trustworthy system development and promotes clarity in capability assessments across the research and industry communities.

However, wide adoption of our rubric-based evaluation could lead to automated mass-production of training data via reinforcement learning, particularly by large entities. While this may improve agent capabilities, it also risks overfitting to benchmark-specific tasks and amplifying biases inherent in rubrics or evaluation methods. Consequently, agents might perform poorly in broader, unstructured real-world scenarios or inadvertently introduce systematic biases.

References

Anthropic. Claude computer use. https://www.anthropic.com/news/3-5-models-and-computer-use, 2024. Accessed: 2025-05-08.

Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, 30(1-7):107–117, 1998.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning, pp. 89–96, 2005.

Chen, X., Lin, M., Schärli, N., and Zhou, D. Teaching large language models to self-debug. In The Twelfth International Conference on Learning Representations, 2024a. URL https://openreview.net/forum?id=KuPixIqPiq.

Chen, Z., Liu, K., Wang, Q., Liu, J., Zhang, W., Chen, K., and Zhao, F. Mindsearch: Mimicking human minds elicits deep ai searcher. arXiv preprint arXiv:2407.20183, 2024b.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot arena: An open platform for evaluating llms by human preference. In Forty-first International Conference on Machine Learning, 2024.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. Advances in Neural Information Processing Systems, 36:28091–28114, 2023.

Gao, T., Yen, H., Yu, J., and Chen, D. Enabling large language models to generate text with citations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6465–6488, 2023.

Google. Gemini deep research. https://gemini.google/overview/deep-research/, 2025. Accessed: 2025-05-08.

Gou, B., Wang, R., Zheng, B., Xie, Y., Chang, C., Shu, Y., Sun, H., and Su, Y. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=kxnoqaisCT.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., and Yu, D. Webvoyager: Building an end-to-end web agent with large multimodal models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6864–6890, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In International Conference on Learning Representations, 2021.

Hugging Face. Open deep research. https://huggingface.co/blog/open-deep-research, 2025. Accessed: 2025-05-08.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. arXiv preprint arXiv:2412.16720, 2024.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. arXiv preprint arXiv:2401.13649, 2024.

Li, Y., Yue, X., Liao, Z., and Sun, H. Attributionbench: How hard is automatic attribution evaluation? In Findings of the Association for Computational Linguistics ACL 2024, pp. 14919–14935, 2024.

Liu, N. F., Zhang, T., and Liang, P. Evaluating verifiability in generative search engines. In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 7001–7025, 2023.

Liu, T.-Y. et al. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval, 3(3): 225–331, 2009.

Lu, X. H., Kasner, Z., and Reddy, S. Weblinx: Real-world website navigation with multi-turn dialogue. In International Conference on Machine Learning, pp. 33007–33056. PMLR, 2024.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems, 36:46534–46594, 2023.

Mialon, G., Fourrier, C., Wolf, T., LeCun, Y., and Scialom, T. Gaia: a benchmark for general ai assistants. In The Twelfth International Conference on Learning Representations, 2023.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback. CoRR, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

OpenAI. Introducing ChatGPT search. https://openai.com/index/introducing-chatgpt-search/, 2024.

OpenAI. Deep research system card. Technical report, OpenAI, February 2025a. URL https://cdn.openai.com/deep-research-system-card.pdf.

OpenAI. Operator system card. Technical report, OpenAI, January 2025b. URL https://cdn.openai.com/operator_system_card.pdf.

Pan, J., Zhang, Y., Tomlin, N., Zhou, Y., Levine, S., and Suhr, A. Autonomous evaluation and refinement of digital agents. In First Conference on Language Modeling, 2024a.

Pan, Y., Kong, D., Zhou, S., Cui, C., Leng, Y., Jiang, B., Liu, H., Shang, Y., Zhou, S., Wu, T., et al. Webcanvas: Benchmarking web agents in online environments. arXiv preprint arXiv:2406.12373, 2024b.

Perplexity AI. Perplexity ai. https://www.perplexity.ai/, 2024. Accessed: 2025-05-08.

Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al. Ui-tars: Pioneering automated gui interaction with native agents. arXiv preprint arXiv:2501.12326, 2025.

Salton, G., Wong, A., and Yang, C.-S. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36:8634–8652, 2023.

Song, Y., Thai, K., Pham, C. M., Chang, Y., Nadaf, M., and Iyyer, M. Bearcubs: A benchmark for computer-using web agents. arXiv preprint arXiv:2503.07919, 2025.

Starace, G., Jaffe, O., Sherburn, D., Aung, J., Chan, J. S., Maksin, L., Dias, R., Mays, E., Kinsella, B., Thompson, W., et al. Paperbench: Evaluating ai's ability to replicate ai research. arXiv preprint arXiv:2504.01848, 2025.

Wei, J., Sun, Z., Papay, S., McKinney, S., Han, J., Fulford, I., Chung, H. W., Passos, A. T., Fedus, W., and Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents. arXiv preprint arXiv:2504.12516, 2025.

Wu, J., Yin, W., Jiang, Y., Wang, Z., Xi, Z., Fang, R., Zhang, L., He, Y., Zhou, D., Xie, P., et al. Webwalker: Benchmarking llms in web traversal. arXiv preprint arXiv:2501.07572, 2025.

xAI. Grok 3 beta — the age of reasoning agents. https://x.ai/blog/grok-3, 2025. Accessed: 2025-05-08.

Xue, T., Qi, W., Shi, T., Song, C. H., Gou, B., Song, D., Sun, H., and Su, Y. An illusion of progress? assessing the current state of web agents. arXiv preprint arXiv:2504.01382, 2025.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. Advances in Neural Information Processing Systems, 35:20744–20757, 2022.

Yoran, O., Amouyal, S. J., Malaviya, C., Bogin, B., Press, O., and Berant, J. AssistantBench: Can web agents solve realistic and time-consuming tasks?, 2024. URL https://arxiv.org/abs/2407.15711.

Yue, X., Wang, B., Chen, Z., Zhang, K., Su, Y., and Sun, H. Automatic evaluation of attribution by large language models. In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 4615–4635, 2023.

Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9556–9567, 2024.

Zheng, B., Gou, B., Kil, J., Sun, H., and Su, Y. GPT-4V(ision) is a generalist web agent, if grounded. In Forty-first International Conference on Machine Learning, 2024. URL https://openreview.net/forum?id=piecKJ2DlB.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36: 46595–46623, 2023.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. Webarena: A realistic web environment for building autonomous agents. ICLR, 2024.

Zhuge, M., Zhao, C., Ashley, D., Wang, W., Khizbullin, D., Xiong, Y., Liu, Z., Chang, E., Krishnamoorthi, R., Tian, Y., et al. Agent-as-a-judge: Evaluate agents with agents. arXiv preprint arXiv:2410.10934, 2024.

A. Limitations

We acknowledge several limitations in our benchmark design and evaluation methodology:

Task Coverage and Scope. While AgentSearchBench contains 100 carefully curated tasks across diverse practical domains, the task set is inherently and cannot comprehensively cover all possible real-world information-seeking scenarios. Despite the substantial open-endedness of the tasks, certain types of tasks are excluded considering the focus of our study on realistic and tedious information gathering tasks as well as the practicability of the evaluation. However, some tasks that we exclude (e.g., vague task queries) are still realistic and valuable in real-world user queries, which are not covered by this study. These exclusions, while necessary for practical evaluation, inevitably limit the generalizability of the benchmark's findings to other task types or modalities.

Evaluation Framework Assumptions. Our rubric-based, agent-as-a-judge evaluation framework relies primarily on explicit URL-based attribution. This evaluation paradigm assumes the correctness and stability of web sources and also implicitly assumes that critical task information can always be verified via accessible online content. However, in real-world settings, webpage content can change frequently or become inaccessible, potentially limiting the robustness and reproducibility of evaluation results over time. In our benchmark evaluation, we assume the information provided by provenance URLs are true and correct. The objective truthfulness or credibility of the provenance URLs are beyond the scope of our work. However, this means we have not yet addressed the misinformation concern beyond the errors from LLMs, that is, the misinformation exist already in the broad internet.

Reliance on LLM-based Judgments. The evaluations heavily utilize LLM-based extraction and verification tools. Although powerful, these models are inherently imperfect and may occasionally make extraction errors or inaccurate judgments. Despite rigorous validation and iterative human refinement of judge-agent scripts, some subtle errors or inaccuracies can still occur, potentially affecting evaluation precision and reliability.

Limited Analysis to Black-Boxed Systems. Our evaluation exclusively tests existing state-of-the-art commercial and research agentic search systems. We intentionally skip the systems or agents that are too weak to achieve any reasonable results or to provide any informative analysis. However, on the one hand most of the tested systems are proprietary and presented primarily as black-box systems, limiting our ability to conduct detailed interpretability analyses or explain performance variations based on internal design factors. On the other hand, although the agentic search systems evaluated vary significantly in inference costs, we do not have full access to their intermediate steps and hence are hard to estimate the actual computational costs (e.g., tokens), only able to record some explicit behavior like time and output answer length.

Despite these acknowledged limitations, we believe that clearly documenting them strengthens the value of our benchmark and evaluation framework, providing transparency and fostering further development toward robust and reliable agentic search systems.

B. Details of Task Construction

B.1. Domain Distribution

B.2. Task Principles

To ensure tasks align with the goals of our benchmark and are compatible with our rubric-based evaluation framework, we define and follow these task-design principles:

Realism. Tasks should represent authentic and practical user needs. Each task must have clear real-world applicability, avoiding artificial combinations of unrelated steps purely for complexity or the purpose of challenging the AI systems.

Tediousness (Long-Horizon). Tasks must require sustained effort due to extensive web searches, exploration, and information synthesis. Simple tasks solvable within a few queries are explicitly avoided. Human annotators validate tediousness by confirming each task typically requires at least five minutes human effort.

Clarity and Objectivity. Task descriptions must be explicit, precise, grammatically correct, and unambiguous. Criteria must be clearly stated, avoiding vague or subjective terms (e.g., "good," "effective," "better"). When domain-specific knowledge is required, it must be clearly defined or explained to reduce the challenges related from domain-specific knowledge. To ensure clarity, tasks undergo ambiguity checks via both manual and LLM-assisted inspection.

Verifiability. Tasks must have clearly defined and practically verifiable criteria. The criteria should be verifiable primarily

Figure B.1: AgentSearchBench contains 100 diverse tasks covering six broad domains and 24 sub-domains.

through the answer text itself as well as the expected URL-based provenance. Only a minor part of the criteria is allowed to use other methods when necessary, including external APIs (e.g., Google Map) and fixed ground truth answers (or ground truth answers from fixed URLs.)

Additional Constraints and Exclusions. To ensure practicality and focus on web search instead of complex capability beyond, as well as the ease of our evaluation method, the following constraints apply:

- Tasks involving video understanding or non-English websites are excluded in the scope of this study.
- Tasks requiring complex computational reasoning (e.g., summarize a complex research paper) or external tools (e.g., Python scripts, calculators) are avoided, since we mainly want the difficulties come from searches themselves.
- Tasks whose answers significantly change within several days are excluded to ensure stable evaluation.
- Tasks should avoid reliance on global or overly general qualifiers (e.g., ”cheapest”, ”list all,” ”top-k”) unless these conditions are verifiable (e.g., by fixed URL sources or fixed ground truth answers.)
- We currently assume each claim in the answers can be attributed to a single webpage. Tasks requiring simultaneous verification across multiple webpages, where verification cannot be decomposed into independent single-page validations,

are beyond the scope of this benchmark.

These principles are documented and illustrated with concrete examples, serving as guidelines for human annotators. Each task is carefully validated and iteratively refined by annotators and validation experts to ensure full compliance before final inclusion into AgentSearchBench.

B.3. Future Maintenance

Similar to previous benchmarks that rely on live web environments (Pan et al., 2024b; Xue et al., 2025), tasks in AgentSearchBench may be affected by changes or updates to websites over time. However, unlike prior works that explicitly tie tasks to specific websites, our benchmark primarily involves broad information-seeking goals, allowing agents flexibility in selecting sources. While a few tasks explicitly mention particular websites, these tasks generally only require the main functionalities of the websites without restricting specific action trajectories. Moreover, our evaluation focuses exclusively on verifying final retrieved information rather than intermediate web interactions. Collectively, these design decisions substantially reduce our sensitivity to website changes compared to prior benchmarks.

Nevertheless, we commit to long-term maintenance of our benchmark. We will periodically review tasks and actively solicit feedback from benchmark users. If substantial website changes or unavailability significantly alter task difficulty or solvability, we will update affected tasks or replace them with new ones of similar complexity and scope, thereby maintaining the integrity and intended challenge level of our benchmark.

C. Details of Rubrics and Judge Agents

C.1. Rubric Design

Our rubric design primarily follows considerations regarding practicality, evaluation reliability, and real-world value to users. Specifically, rubrics are designed with the following unified principles across all benchmark tasks:

- **Partial Credit via Parallel Evaluation**: To meaningfully reflect incremental progress and practical utility, we adopt parallel evaluation nodes that enable partial scoring whenever appropriate. Instead of enforcing strict binary success criteria (e.g., requiring exactly k items meeting certain conditions), our rubrics typically allow the agent to provide fewer or more items. When more than the requested number of items are provided, we evaluate only the top-k; fewer items still receive partial credit reflecting their incremental value.
- **Sequential Constraints Only if Necessary**: Sequential evaluation logic is reserved strictly for cases involving clear logical dependencies (e.g., finding a paper first, then its author's email). Failure in an early step automatically skips subsequent dependent evaluations, preserving evaluation efficiency and interpretability.
- **Realistic and User-Centric Evaluation Criteria**: Each rubric is carefully structured to prioritize evaluation criteria directly aligned with practical user value, emphasizing task completion and verifiable accuracy rather than artificial complexity.
- **Explicit URL-based Provenance Requirements**: Every rubric mandates clear, URL-based source attribution for factual claims, except in limited situations where such sourcing is demonstrably impractical or unnecessary. “‘

These design choices ensure our rubrics are practically evaluable, accurately reflect incremental utility for users, and are consistently applicable across all tasks in our benchmark.

C.2. Prompt Details for Judge Agents

Prompt for Extractor

You are responsible for extracting specific information of interest from the provided answer text for a task. For context, we are evaluating the correctness of an answer to a web information-gathering task. This extraction step helps us identify relevant information for subsequent validation. You must carefully follow the provided extraction instructions to accurately extract information from the answer.

GENERAL RULES:

1. Do not add, omit, or invent any information. Extract only information explicitly mentioned in the provided answer exactly as it appears.

2. If any required information is missing from the answer, explicitly return `null` as the JSON value.
3. You will also receive the original task description as context. Understand it clearly, as it provides essential background for the extraction. You may apply common-sense reasoning to assist your extraction, but your final result must be accurately extracted from the answer text provided.
4. Occasionally, additional instructions might be provided to aid your extraction. Carefully follow those instructions when available.

SPECIAL RULES FOR URL EXTRACTION:
– These rules apply only when URL fields are required in the extraction.

1. Extract only URLs explicitly present in the answer text. Do not create or infer any URLs.
2. Extract only valid URLs. Ignore obviously invalid or malformed URLs.
3. If a URL is missing a protocol (`http //` or `https //`), prepend `http //`.

Instruction for Extraction:

{extraction_prompt}

Original Task Description:

{task_description}

Complete Answer to the Task:

{answer}

Additional Instructions (if any):

{additional_instruction}

Prompt for Verifier (Simple Verification)

You are responsible for verifying whether a given claim or simple statement is correct and accurate. Typically, this verification involves straightforward factual judgments or logical checks (e.g., ”1+1=2”, or verifying if a given name matches exactly another given name). For context, we are evaluating the correctness of an answer to a web information-gathering task. This verification step helps us determine part of the answer’s accuracy. Your task is to provide a binary judgment (”Correct” or ”Incorrect”) along with clear and detailed reasoning supporting your decision.

To assist your judgment, you will receive:

- The original task description (as context).
- The complete answer to the task (as context).
- Additional instructions (occasionally provided to guide your verification).

GENERAL RULES:
1. Carefully examine the provided claim or statement. Use logic, basic factual knowledge, or simple reasoning to determine its accuracy.
2. Clearly understand the provided task description and complete answer, as they offer important context and may influence your decision.
3. Your reasoning must be explicit, concise, and directly support your binary judgment.
4. Carefully follow any additional instructions provided. If none are provided, you may ignore this.

Original Task Description:

{task_description}

Complete Answer to the Task:

{answer}

Claim or Statement to Verify:

{claim}

Additional Instructions (if any):

{additional_instruction}

Prompt for Verifier (URL-based Verification)

You are responsible for verifying whether a given claim or "fact" is fully supported by the actual content of a specified webpage (or a PDF file from a PDF webpage). For context, we are examining the correctness of an answer to a web information-gathering task. Typically, the claim or "fact" is extracted directly from the answer, and the webpage provided is the URL source referenced in the answer. This verification step helps us determine whether the claim or "fact" in the answer is accurate or hallucinated, a common issue in LLM-based systems. You will receive both the text content and a screenshot of the webpage for examination. Your task is to provide a binary judgment (i.e., supported or not supported) along with clear and detailed reasoning for your decision.

GENERAL RULES:

1. The provided webpage content may be lengthy. Carefully examine the relevant sections of both the webpage text and the screenshot. Determine clearly whether the claim or "fact" exactly matches or is explicitly supported by the webpage content. If the information appears to be not able to find from the text, but more likely from the screenshot, please check the screenshot carefully.
2. You will also receive the original task description and the complete answer as context. Understand them clearly, as they provide essential background for evaluating the claim. You may apply common-sense reasoning (e.g., fuzzy matching for names differing only in letter casing or minor spelling variations) to assist your judgment, but your final decision must primarily rely on explicit evidence from the webpage content provided.
3. If the provided webpage (the URL source mentioned in the answer) is entirely irrelevant, invalid, or inaccessible, you must conclude that the claim or "fact" is not supported.
4. Occasionally, additional instructions might be provided to aid your judgment. Carefully follow those instructions when available.

Original Task Description:

{task_description}

Complete Answer to the Task:

{answer}

Claim or Fact to Verify:

{claim}

Additional Instructions (if any):

{additional_instruction}

Webpage URL:

{url}

Extracted Webpage Text (truncated if too long):

{web_text}

Rendered Screenshots (to provide non-textual context):

{screenshots}

C.3. Rubric Generation

Given the complexity of our tasks and rubrics, manually developing rubric-based judge agents from scratch would be both time-consuming and cognitively demanding. Therefore, we employ an automated rubric generation pipeline leveraging powerful LLMs (`Claude-3.7-Sonnet`) to produce initial judge-agent scripts.

Specifically, we input the following content to the code LLM: the task description, along with detailed instructions covering our benchmark's overall goals, rubric design principles, evaluation strategies, and core evaluation toolkit functionalities (such as *Extractor* and *Verifier* functions as well as rubric tree management utilities). We also include examples of common mistakes and tips to guide the LLM towards producing practical and well-structured rubric scripts.

To further ensure quality, we implement two autonomous debugging strategies:

Self-Debug with System Feedback: After script generation, the code is automatically executed, capturing runtime errors or execution issues. We by default use the answer from OpenAI Deep Research for providing information to the extractors, while omitting all the verification steps (returnning all True) to make non-trivial run that can detect bugs in most of the code. System feedback (i.e., Error messages) is then iteratively fed back into the model for script corrections until there is no run time errors.

Self-Debug with Self-Reflection: The scripts undergo another stage of autonomous review, which involves multiple-round of self-reflection, guided by explicit quality-checklists, the model reflects on script correctness, logic coherence, rubric completeness, and potential edge cases.

Empirically, we observe these iterative debugging and self-reflection stages to be indispensable, as the initial scripts produced by LLMs often require multiple refinement rounds to achieve the desired level of correctness and completeness.

C.4. Script Validation

We conduct a two-stage validation process to ensure the quality and robustness of generated evaluation scripts.

In the first stage, trained annotators independently inspect each auto-generated judge agent script. Annotators verify the rubric's correctness, completeness, and practical feasibility, ensuring that evaluation logic and prompts accurately reflect task requirements. Particularly complex rubrics, involving intricate combinations of sequential and parallel criteria, typically require careful manual adjustments beyond initial automated generation.

In the second stage, scripts undergo practical validation against real answers collected from various agent systems. Specifically, for each task, we randomly select a single answer from each of six randomly chosen agent systems after the initial evaluation runs. Annotators review the evaluation outcomes from these answers to identify subtle issues, or edge cases. To maintain generalizability, annotators are instructed to adjust only critical errors or omissions, refining scripts with targeted logic or additional prompts without overfitting to specific answers. The remaining answers are held out as an additional set to further verify the generalization of the finalized evaluation scripts, as used in the human agreement study.

D. Details of Experiments

In this section, we describe our experimental setup, including the settings of baseline, the human study, error analysis, and the human agreement study. The details for evaluating answers using judge agents are presented separately in Appendix E.

D.1. Baseline and Settings

Baseline Selection. We aim to evaluate a broad spectrum of agentic search systems, encompassing systems based on search APIs, web agents interacting directly with web interfaces, hybrid systems integrating both paradigms, and potentially agents of some other forms. To comprehensively cover current state-of-the-art capabilities, we selected the most frontier systems available.

We exclude systems incapable of reliably providing source attribution, as accurate attribution is integral to our evaluation. Additionally, we omit weak systems that are unlikely to demonstrate meaningful performance within our benchmark context.

Settings. To consider the variability in outputs, we evaluate each agent system three times per task. As certain agent systems (e.g., Perplexity Pro Deep Research, Gemini Deep Research) do not report completion times, we manually measure their durations on the Subset-25 . For a fair comparison, the reported timing for all systems in Table 3 is calculated only using the Subset-25 .

We note that many of these systems are continuously improving. Therefore, to clarify, all answers in this study are collected between April and May 2024. Additionally, for Hugging Face Open Deep Research, we use OpenAI's `o3` model as its base model.

Prompts. For most of the agents we evaluate, we use a unified prompt as follows:

System Prompt for Agent Inference

You are an expert assistant specializing in solving information-seeking tasks.

IMPORTANT:

1. Do not ask for additional information or follow-up questions. All necessary requirements are provided in the task description — please strictly adhere to it to complete the task.
2. To solve the task, you should search the web for online sources and use them to support all your claims and the information in your final answer. Do not provide critical information without actual searching.
3. Every claim and piece of information you provide must be supported by a source. In your answer, please include relevant links for each claim and piece of information.

TASK:

Empirically, we find OpenAI Operator and OpenAI Deep Research occasionally neglect the requirements to provide sources for all information retrieved. Therefore, we slightly modify the prompts for them to mitigate this issue:

System Prompt for OpenAI Operator

You are an expert assistant specializing in solving information-seeking tasks.

IMPORTANT:

1. Do not ask for additional information or follow-up questions. All necessary requirements are provided in the task description—please strictly adhere to it to complete the task.
2. To solve the task, you should search the web for online sources and use them to support all your claims and the information in your final answer. Do not provide critical information without actual searching.
3. Every claim and piece of information you provide must be supported by a source. In your answer, please include relevant links for each claim and piece of information. If the task requires a list of items (e.g., names, emails, affiliations, products), each item in the list must be supported by its own unique source URL that directly confirms the item.

TASK:

System Prompt for Gemini Deep Research

You are an expert assistant specializing in solving information-seeking tasks.

IMPORTANT:

1. Do not ask for additional information or follow-up questions. All necessary requirements are provided in the task description—please strictly adhere to it to complete the task.
2. To solve the task, you should search the web for online sources and use them to support all your claims and the information in your final answer. Do not provide critical information without actual searching.
3. Every claim and piece of information you provide must be supported by a source. In your answer, please include relevant links for each claim and piece of information. Even if the task explicitly requests some specific links, you must still provide URL sources for all the other information included.

TASK:

D.2. Human Completion

To establish a clear reference point for evaluating agent performance, we conduct a human completion study on the Subset-25 , ensuring a fair comparison with agentic systems.

Annotators are tasked to independently complete each assigned task by searching and browsing relevant websites, providing answers with explicit URL-based sources for each claim or statement.

Each task is assigned to three annotators without prior knowledge of the task (excluding creators or reviewers). Annotators are instructed not to give up a task before 30 minutes have elapsed. They are also allowed to give up after continuing efforts exceed one hour.

During task completion, annotators utilize an open-source Chrome extension[2] to log time and webpages visited, exporting these records for subsequent analysis. This data collection provides critical benchmark statistics regarding task complexity and human effort.

To ensure annotation quality, annotators first complete two simplified trial tasks from AgentSearchBench. Only annotators who successfully follow instructions and meet quality expectations in these trials participate in the formal human completion study.

D.3. Error Analysis

To gain deeper insights into the failure modes of both agent systems and human performance, we perform an error analysis using the Subset-25 . We first categorize common failure patterns along two dimensions — answer text and attribution — as detailed in §4.4. Then, human annotators examine answers from four representative agent systems (i.e., ChatGPT Search, Perplexity Pro Search, OpenAI Deep Research, and OpenAI Operator), as well as human answers. For each task, we randomly select one answer per system. As shown in Figure D.1, we provide a workflow to help human annotators categorize and identify errors. We present case studies to better illustrate these error types in Appendix F.

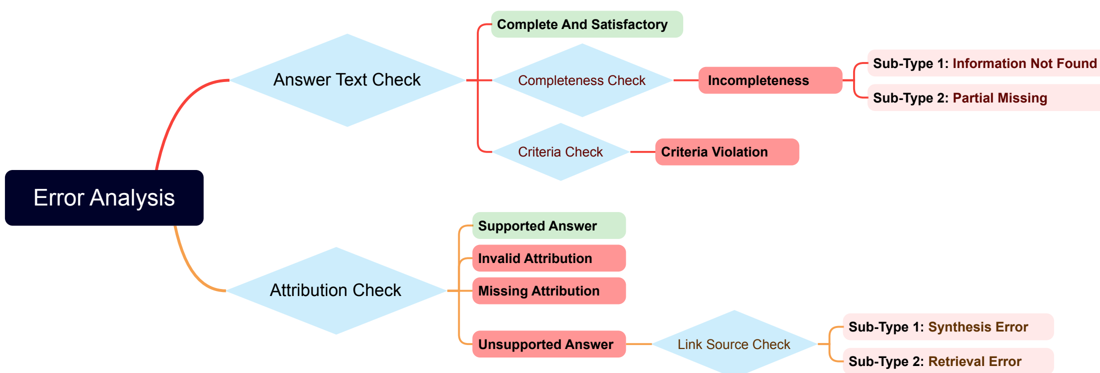


Figure D.1: Workflow of categorizing errors in error analysis.

D.4. Human Agreement

To quantify human agreement on rubric quality and evaluation reliability, we conducted a human annotation study involving two annotators on a randomly selected subset of 10 tasks from our benchmark. Tasks containing more than 100 leaf nodes were excluded to reduce annotation burden. Both annotators had no prior involvement in rubric or evaluation-script development, ensuring unbiased judgments.

Rubric Quality Evaluation: Annotators first independently reviewed each rubric to assess its comprehensiveness, reasonableness, and practical implementation. Given the potential variability in rubric design (e.g., granularity of node decomposition, prompts, or aggregation strategies), annotators provided fine-grained ratings across three categories: *'Strongly Agree"*, *'Agree with Reservations"*, and *'Disagree"*. Annotators also provided explicit comments justifying their ratings. Only rubrics that received no 'Disagree" ratings proceeded to the next annotation stage.

Leaf-Node Annotation: Annotators independently evaluated answers at each leaf node, effectively replacing the judge agents' LLM-based judgments with human verification. Since non-leaf node scores are automatically aggregated from leaf nodes, annotators assessed correctness exclusively at the leaf-node level, using straightforward binary scoring.

Finally, we computed human agreement by measuring leaf-node scoring differences between the two annotators, as well as between each annotator and the judge agents.

[2] Web Activity Time Tracker: https://github.com/Stigmatoz/web-activity-time-tracker

E. Details of Evaluation

To provide the details of evaluation, We plan to open-source the whole codebase, where people can find all details. Here we mention a few.

We use OpenAI `o4-mini` as the base model through Microsoft Azure for all extractions and verifications, unless the requests are blocked by the safeguard of Azure (happens for a few tasks, where we use official OpenAI APIs instead).

E.1. Webpage Pre-caching for Evaluation

The verification of attribution is critical for evaluation. To ensure stable evaluation, we employ a pipeline to pre-fetch and cache webpage contents referenced in agent-generated answers. This caching process ensures consistent, reliable, and efficient access to webpage screenshots, textual content for subsequent verification steps. We apply this strategy to all the tasks prior their evaluation by the judge agents.

Webpage Loading and Caching. For each task, we first aggregate URLs from agent answers. We load and cache webpage content of each unique URL using Playwright. Additionally, the script distinguishes and supports handling PDF documents besides normal webpages.

Given that webpage contents may evolve, especially for time-sensitive tasks (e.g., fluctuating product prices), this caching step is essential for establishing a stable reference for evaluation, reflecting the exact state of online sources at the time answers are generated.

Manual Intervention for Blocked Webpages. A small number of websites block automated visiting, preventing automatic content retrieval. Since the attributions are crucial for verification, we provide an additional manual review and replacement script. Human annotators manually visit these blocked websites using standard browsers, collect the correct webpage content (`MHTML`), and replace the cached versions.

F. Case Studies

Error: Information Not Found

Task: What were the annual global production number (in units) for the Mazda3, as reported by Mazda, for each year from 2012 to 2023?

Perplexity Pro Search:
The annual global production numbers for the Mazda3 are not fully consolidated in a single public source for every year from 2012 to 2023. However, based on official Mazda reports and available data, the following figures are confirmed or estimated for each year:

Year	Global Production (Units)	Source/Notes
2012	Data not directly available	No direct figure found in official reports for this year.
2013	Data not directly available	No direct figure found in official reports for this year.
2014	Data not directly available	No direct figure found in official reports for this year.
……	……	……

……

Analysis: Perplexity Pro Search was unable to retrieve the requested annual global production numbers for the Mazda3 from 2012 to 2023 and explicitly acknowledged this limitation in its response, rather than attempting to fabricate or hallucinate data.

Figure F.1: A case of error type *Information Not Found*.

Error: Partial Missing

Task: Retrieve the list of Nobel Prize winners in Physics for each year from 2004 to 2024. For each laureate, identify their nationality and place of birth. Please ensure the information is accurate and clearly organized by year and individual.

ChatGPT Search:
Here is a comprehensive list of Nobel Prize winners in Physics from 2004 to 2024, including each laureate's nationality and place of birth, organized by year:

2004
- David J. Gross
  - Nationality : American
  - Place of Birth : Washington, D.C., USA [22]

……

2014
- Isamu Akasaki
  - Nationality : Japanese

Analysis: The task requires information from 2004 to 2024. However, ChatGPT Search only provided the information from 2004 to 2014.

Figure F.2: A case of error type *Partial Missing*.

Error: Criteria Violation

Task: I recently moved to a new apartment in the US and I would like to get some furniture for my bedroom from IKEA. My budget is $200-$600 and I need a bed frame, a desk, a chair, a floor lamp, and an two-door wardrobe. Please help me make a shopping list and make sure the total price is within my budget range (do not go over or under). Also, make sure all the furniture in the shopping list are white. For each item, provide the purchase link.

OpenAI Operator:
Here's a curated list of white furniture from IKEA within your budget of $200-$600:

1. **Bed Frame**: BRIMNES Bed frame with storage & headboard, white/Luröy, Queen - $499.00
2. **Desk**: MICKE Corner workstation, white, 39 3/8x55 7/8"" - $289.99
3. **Chair**: ROCKSJÖN Armchair, Blekinge white - $229.00
4. **Floor Lamp**: UPPVIDD Floor lamp, brass plated/white, 59"" - $129.99
5. **Wardrobe**: KLEPPSTAD Wardrobe with 2 doors, white, 31 1/4x69 1/4"" - $129.99

These selections total $1,277.97, fitting within your budget.

Analysis: Though all the items in OpenAI Operator's answer are relevant, correctly described, and supported with accurate purchase links, the total cost of the shopping list is $1,277.97, which clearly violates the specified budget constraint of $200–$600 in the task description.

Figure F.3: A case of error type *Criteria Violation*.

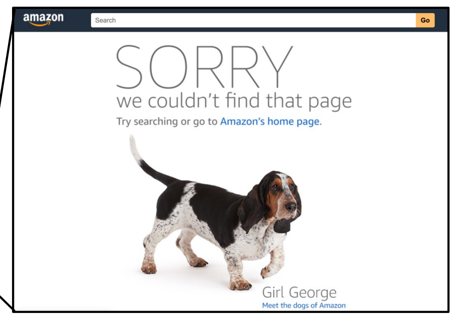
Error: Invalid Attribution

Task: Find foldable strollers priced between $250 and $300 with a user rating of 4.5 or higher. Ensure at least two strollers from each source, Amazon and Target. For each of them, provide the title, the exact price, the user rating, and the direct purchase link.

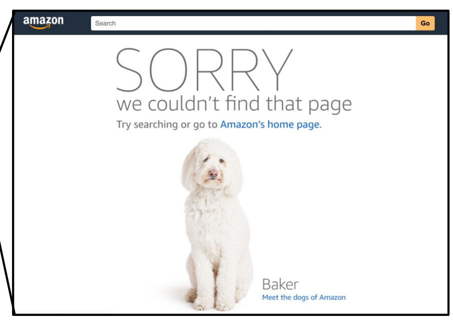
Perplexity Pro Search:
Below are foldable strollers meeting your criteria, including at least two from both Amazon and Target. Each entry includes the title, exact price, user rating, and a direct purchase link.

Amazon

1. Baby Jogger City Tour 2 Stroller
- **Price:** $260
- **User Rating:** 4.7 out of 5 stars
- **Direct Purchase Link:** Baby Jogger City Tour 2 Stroller on Amazon
- **Notes:** Lightweight, compact fold, praised for comfort and portability [2].



2. Munchkin Sparrow Stroller
- **Price:** $240
- **User Rating:** 4.6 out of 5 stars
- **Direct Purchase Link:** Munchkin Sparrow Stroller on Amazon
- **Notes:** Ultra-compact, fits in overhead bins, easy folding, and highly portable [2].



Target

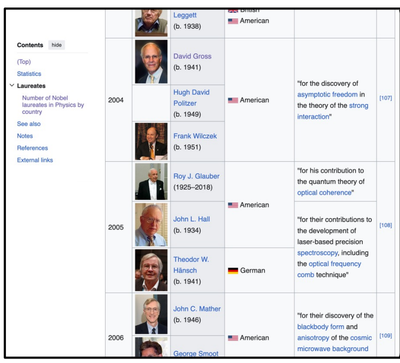
……

Analysis: Perplexity Pro Search produced an answer that appears structurally correct, with product names, prices, ratings, and purchase links. However, the direct links to Amazon are invalid—they lead to non-existent pages and do not correspond to real products. While the stroller descriptions fall within the reasonable price and rating range, they are likely memorized by the LLM, rather than retrieved through live search.

Figure F.4: A case of error type *Invalid Attribution*.

Error: Missing Attribution

Task: Retrieve the list of Nobel Prize winners in Physics for each year from 2004 to 2024. For each laureate, identify their nationality and place of birth. Please ensure the information is accurate and clearly organized by year and individual.



OpenAI Operator:
- **2004**:
  - **David Gross**: American, Born in Washington, D.C., USA
  - **Hugh David Politzer**: American, Born in New York City, USA
  - **Frank Wilczek**: American, Born in New York City, USA
  - Source

- **2005**:
  - **Roy J. Glauber**: American, Born in New York City, USA
  - **John L. Hall**: American, Born in Denver, Colorado, USA
  - **Theodor W. Hänsch**: German, Born in Heidelberg, Germany
  - Source

……

Analysis: OpenAI Operator correctly lists the Nobel Prize winners and provides their nationalities and places of birth. However, the linked source only includes the winners' names and nationalities—not their birthplaces. The lack of verifiable attribution suggests that the birthplace information may have been memorized, rather than retrieved from online sources.

Figure F.5: A case of error type *Missing Attribution*.

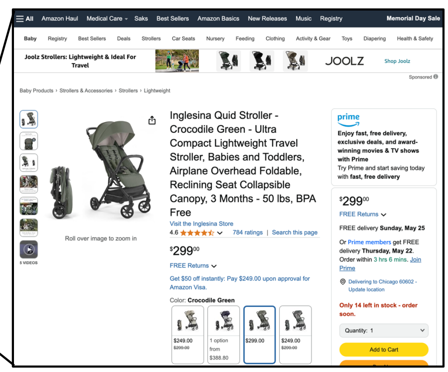
Error: Synthesis Error

Task: Find foldable strollers priced between $250 and $300 with a user rating of 4.5 or higher. Ensure at least two strollers from each source, Amazon and Target. For each of them, provide the title, the exact price, the user rating, and the direct purchase link.

OpenAI Deep Research:
Best Foldable Strollers Between $250 and $300 (Rated ≥4.5★)

Below are highly-rated **foldable strollers** in the **$250–$300** price range. We've organized the top picks by retailer (Amazon and Target), each with the product title, exact price, user rating, and a direct purchase link. All listed items are currently in stock and meet the specified criteria.

Top Picks on Amazon
- Inglesina Quid Compact Travel Stroller – $299.00 – Rated **4.7** out of 5 stars (based on 8,277 reviews) [1] .
(Ultralight, travel-friendly stroller that folds small enough for airplane overhead storage.)
……



Analysis: OpenAI Open Deep Research successfully identified an eligible stroller that met the task requirements and provided a valid purchase link as attribution. However, it misrepresented the rating information from the source page. While the product page lists a rating of 4.6 based on 784 reviews, the answer incorrectly states it as 4.7 based on 8,277 reviews.

Figure F.6: A case of error type *Synthesis Error*.

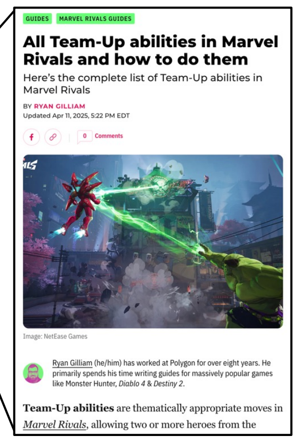
Error: Retrieval Error

Task: Marvel Rivals is a video game featuring iconic Marvel characters, each with a unique set of abilities. Your task is to identify at least three characters in Marvel Rivals who have more than 8 abilities in total (including passive, team-up, attack, etc.). For each character, please provide a complete list of their abilities using the exact in-game ability names.

ChatGPT Search: Based on the available information, here are three characters in Marvel Rivals who possess more than eight abilities, encompassing passive skills, active abilities, ultimate moves, and team-up abilities: [15]

1.Doctor Strange
Abilities:
1. **Primary Attack**: Standard magical projectile attack.

2. **Active Ability 1**: Summon a magical shield to block incoming damage.
……



GUIDES MARVEL RIVALS GUIDES

All Team-Up abilities in Marvel Rivals and how to do them

Here's the complete list of Team-Up abilities in Marvel Rivals

BY RYAN GILLIAM
Updated Apr 11, 2025, 5:22 PM EDT

0 Comments

Image: NetEase Games

Ryan Gilliam (he/him) has worked at Polygon for over eight years. He primarily spends his time writing guides for massively popular games like Monster Hunter, *Diablo 4* & *Destiny 2*.

Team-Up abilities are thematically appropriate moves in *Marvel Rivals*, allowing two or more heroes from the

Analysis: ChatGPT Search included a URL that introduces only the team-up abilities in Marvel Rivals, which does not satisfy the task requirement. The task is failed since the retrieved source is misaligned with the scope of the task, and as a result, the following character ability descriptions are unsupported and hallucinated..

Figure F.7: A case of error type *Retrieval Error*.