Lost in the Maze: Overcoming Context Limitations in Long-Horizon Agentic Search

Anonymous Authors[1]

Abstract

Long-horizon agentic search requires iteratively exploring the web over long trajectories and synthesizing information across many sources, enabling powerful applications like deep research systems. In this work, we show that popular agentic search frameworks struggle to scale to long trajectories primarily due to context limitations—they accumulate long, noisy content, hit context window and tool budgets, or stop early. Then, we introduce SLIM (**S**imple **L**ightweight **I**nformation **M**anagement), a simple framework that separates retrieval into distinct search and browse tools, and periodically summarizes the trajectory, keeping context concise while enabling longer, more focused searches. Across a wide range of long-horizon tasks, SLIM achieves comparable performance at substantially lower cost and far fewer tool calls than strong open-source frameworks with both proprietary and open-weight models, including RL-trained models for deep research. Specifically, with o3 as the base model, SLIM achieves 56% on BrowseComp and 33% on HLE, outperforming all open-source frameworks by 8 and 6 absolute points, respectively, while incurring 4–6x fewer tool calls. With GLM-4.7 Flash, SLIM achieves 10 points improvement over the next best open-source framework, SEARCH-O1, on BrowseComp using a third of the cost. To systematically understand failure modes in long-horizon agentic search, we develop an automated fine-grained trajectory analysis pipeline and error taxonomy, and find that SLIM exhibits fewer hallucinations than prior systems. We hope our analysis framework and simple tool design inform future long-horizon agents[1].

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

Preliminary work. Under review at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Do not distribute.

[1]Our code will be made publicly available.

1. Introduction

Long-horizon agentic search involves performing searches over long trajectories and reasoning over many sources, and requires powerful systems that can explore diverse sources and leverage tools effectively. The ability to reason over long trajectories serves as the foundation for exciting applications such as deep research (OpenAI, 2025; Google, 2025; xAI, 2025). Due to its immense potential in solving complex tasks, long-horizon systems have been a key focus in the community, eliciting the development of many proprietary and open-source frameworks. Among open-source systems, HuggingFace Open Deep Research (Roucher et al., 2025) and GPT Researcher (Elovic, 2023) opt for complex multi-agent orchestration while SEARCH-O1 (Li et al., 2025b) uses a single agent. However, despite the numerous approaches, they still fail in complex long-trajectory settings, and there are no systematic approaches to analyze their trajectories and identify the failure modes.

In this work, we first analyze existing frameworks by examining their trajectory outcomes on BrowseComp (Wei et al., 2025), a challenging long-horizon agentic search benchmark. Our analysis shows that these frameworks still struggle with long-trajectory tasks, failing on more than 50% of the samples—most of the failures are due to hitting the context window limit, running out of tool budget, or stopping too early. We attribute these failure modes to poor context management that can fill the context window with noisy information that derails long search trajectories. The limited context restricts the number of turns in each trajectory, resulting in incomplete information gathering. To overcome these limitations, we design SLIM (**S**imple **L**ightweight **I**nformation **M**anagement), a framework with three simple yet powerful components—search, browse, and summarization—that effectively manage the context size of long-horizon systems. The simple tool design allows LLMs to interleave searching for diverse information and browsing promising pages without spending unnecessary tool calls on noisy results. Furthermore, the summarization module acts as a general-purpose context manager that can reduce long trajectories into more condensed summaries. These design choices combine to enable scaling to longer trajectories while maintaining a concise context and reduced

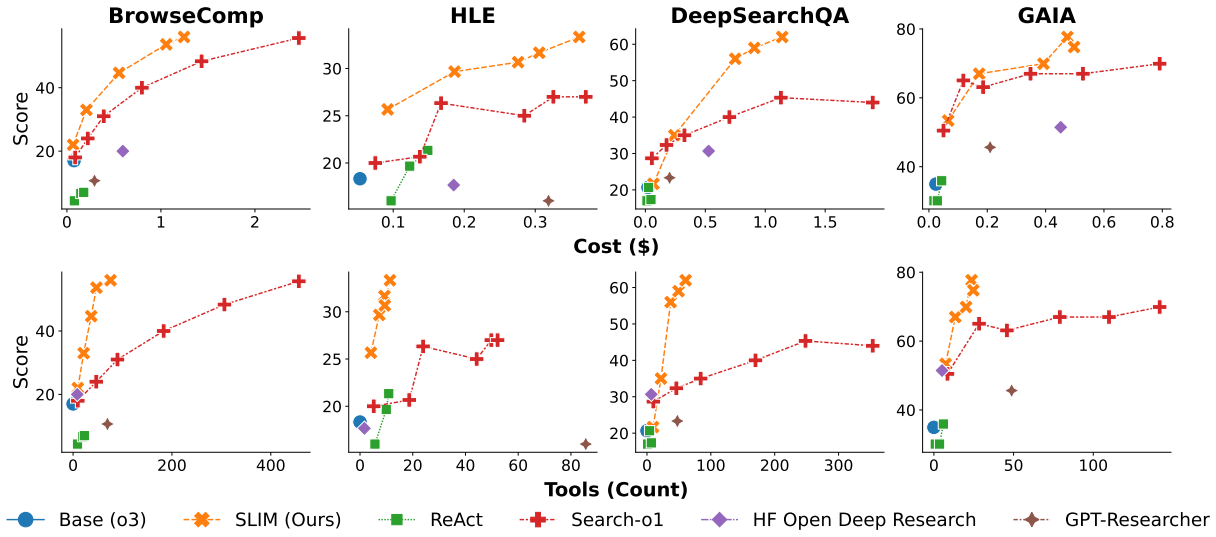

Figure 1. With o3 as the base model, SLIM achieves better performance than existing frameworks on across all datasets while using more than 4-6x fewer tool calls and lower overall costs, which account for LLM token usage and tool costs.

tool costs. Under a comparable cost budget, with o3 as the base model, SLIM significantly outperforms the previous best open-source frameworks across all benchmarks, while requiring only 15-25% of the tool calls (Figure 1).

Finally, we develop an automated trajectory-level analysis pipeline that provides fine-grained insights into long-horizon frameworks. To characterize mistakes made by these systems, we develop an error taxonomy identifying common failure modes. Our analysis reveals that SLIM's advantage stems from its robustness to failure modes such as hallucinations and unfocused and generic searches. We hope our analysis pipeline, error taxonomy, and careful design choices in SLIM can serve as a foundation for understanding and improving long-horizon agentic search systems.

2. Preliminaries: Long-Horizon Agentic Search

Previous search tasks can be mostly solved with static retrieval-augmented generation (RAG) systems that leverage at most a few retrieval steps (Lewis et al., 2020; Izacard et al., 2023; Shi et al., 2024), and do not showcase the challenges of realistic, long-horizon agentic search settings. In contrast, we study long-horizon tasks with complex queries that require extensive searches to gather the necessary information and reasoning over different sources to synthesize the answer. In this section, we formalize the task, describe the datasets for studying long-horizon agentic search, and review some previous long-horizon systems.

2.1. Task Formulation

We formalize long-horizon agentic search tasks as follows: given a query q, a corpus of documents $\mathcal{D}$, the system needs to perform a sequence of tool calls to find relevant information from $\mathcal{D}$ and output a final answer o, which is checked against the annotated groundtruth answer a. A critical component of the system is the design of its tools and how it interacts with the corpus; each tool is a function $\mathcal{T}_i(x) \rightarrow y$ that maps arbitrary system-generated inputs x to arbitrary outputs y. Furthermore, agentic systems are often controlled by a tool budget T, the maximum number of tool calls they are allowed to use in any trajectory. The tool budget T also corresponds to the maximum number of turns in a trajectory, as each turn corresponds to one tool call[2]. Thus, how to manage the input context to the underlying LLM across many tool uses and turns is another critical design choice in long-horizon systems. Finally, the final step where the system outputs its final answer does not count towards the tool budget.

In long-horizon agentic search settings, the web is most often used as the corpus $\mathcal{D}$ due to the diversity and complexity of the queries, and each document $d_i = (u_i, t_i, c_i)$ comprises a URL, title, and content. In practice, long-horizon systems typically use search engines $\mathcal{R}(q) \rightarrow \{(u_i, t_i)\}_1^n$ to obtain a list of n web pages with their titles and URLs most relevant to the search query q. Furthermore, a scraping operation $\mathcal{C}(u_i) \rightarrow c_i$ is necessary to obtain the full content of any URL as search engines only provide a list of URLs, but scraping is slow and noisy in practice. In traditional

[2]Some architectures, such as the CodeAgent (Wang et al., 2024) used in HF-ODR, allow for parallel tool calls in one step, but we found that the models we tested do not use this capability.

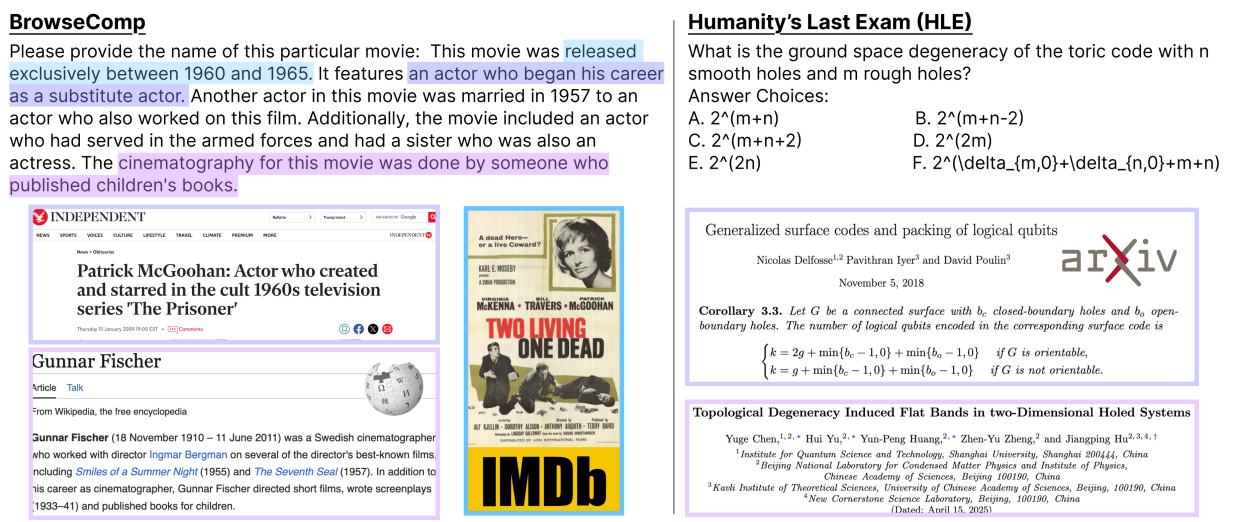

Figure 2. Example queries and their relevant documents for BrowseComp (Wei et al., 2025) and HLE (Phan et al., 2025).

Framework	Architecture	# Tools	Tools	Input to LLM Context	Summarization
REACT	Single-agent	1	Retrieval	All search results	-
SEARCH-O1	Single-agent	1	Retrieval	All search results	Retrieved content
HF-ODR	Multi-agent	11	Search, Browse, Python, ...	Selected search results	Search agent result
GPT-R	Multi-agent	1	Retrieval	All search results	Retrieved content
SLIM (ours)	Single-agent	2	Search, Browse	Selected search results	Task trajectory

Table 1. Comparison of SLIM with existing frameworks. In contrast to single-agent works that bundle search and browsing search results into *one* retrieval tool, we separate it into two distinct tools.

QA settings, since the retrieval tool only needs to be called once due to the simplicity of the queries and the small size of the corpus (i.e., Wikipedia), retrieval returns the full list of documents and their contents $\mathcal{R}_{\texttt{wiki}}(q) \rightarrow \{(t_i, c_i)\}_1^n$. As a result, many long-horizon systems follow a similar design, where the retrieval tool is a single search engine call followed by scraping all returned URLs. However, the complexity of long-horizon agentic search requires many tool calls to gather the necessary information (Li et al., 2025b; Jin et al., 2025b). As we demonstrate empirically later, this naive tool design leads to severe context limitations, where the system is overwhelmed by long, noisy content, motivating the design of more efficient tool interfaces for long-horizon systems.

2.2. Datasets

We select five diverse datasets with naturally difficult queries that require long-trajectory searches and verifiable answers, which ensures the reliability of subsequent analyses. For evaluation, we sample a random subset of 300 instances from each dataset due to the high costs of running long-horizon systems.[3]. Examples are shown in Figure 2.

[3]For GAIA, we evaluate on the subset of 103 text only queries, following Liu et al. (2025b).

BrowseComp (Wei et al., 2025) and GAIA (Mialon et al., 2023) contain challenging queries that test the ability to exhaustively search the web over long trajectories. Humanity's Last Exam (Phan et al., 2025) tests across multiple domains and often requires domain-specific knowledge and applying web information to reasoning-heavy problems. DeepSearchQA (Gupta et al., 2026) and HealthBench (Arora et al., 2025) evaluate long-form generation capabilities across board domains.

2.3. Existing Approaches

We briefly describe some popular approaches to agentic search, ranging from simple single-LLM frameworks to complex multi-agent systems. We summarize the differences between these frameworks in Table 1; more details are in §A.1. For REACT and SEARCH-O1, we vary the tool budget, and for HF-ODR and GPT-R, we follow the default settings.

REACT (Yao et al., 2023) is a simple framework that allows an LLM agent to alternate between thinking and acting, allowing tool calling across many turns. Following the original work, LLM uses a single retrieval tool—given a query, the tool returns a list of top 10 results along with their web contents. All results are then concatenated to the

Table 2. Categorization of different search outcomes and their descriptions.

Outcome	Description
Correct	The system outputs the correct answer
Exceed context	The system exceeds LLM's context window, falling back to not using any tools
Exceed budget	The system exceeds the tool calling or iteration budget
Early stopping	The system outputs an incorrect answer before reaching the iteration budget
No tool used	A special case of early stopping where the system does not use any tools
Misc. error	Due to uncontrollable factors (e.g., API content filters) the system outputs an error

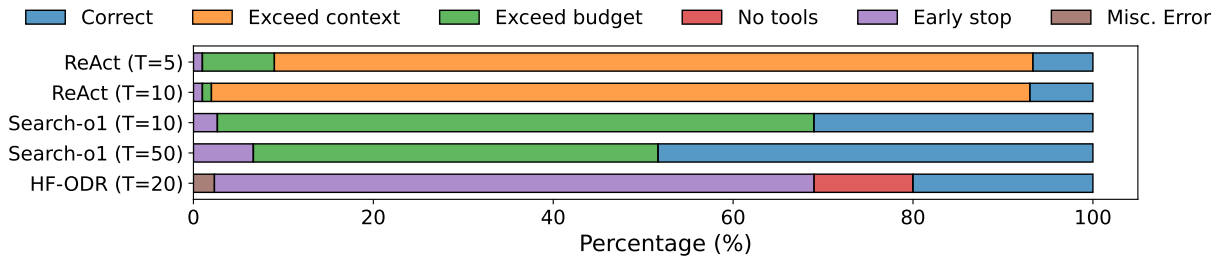

Figure 3. Each framework exhibits distinct outcome trends—REACT predominantly runs out of context window, while SEARCH-O1 is often limited by the tool budget (T). We exclude GPT-R due to its predefined workflow—the outcome can only be correct or incorrect.

agent's context for subsequent steps. **SEARCH-O1** (Li et al., 2025b) builds upon REACT with an additional "reason-in-document" step, where an LLM summarizes the search results and their contents before appending the results to the agent's input context. Although summarization reduces context length for the main LLM compared to REACT, this approach still uses many scraping operations in each search step (one for each result), and summarization incurs a large amount of token usage. **HuggingFace OpenDeepResearch** (HF-ODR; Roucher et al., 2025) leverages a hierarchical structure consisting of a manager agent and a search agent, which iteratively interacts with a search engine, a browser, and other tools (detailed in §A.2). **GPT-Researcher** (GPT-R; Elovic, 2023) is a complex multi-agent system where each agent has distinct roles: research conductor, report generator, context manager, and source curator. The system uses a deep researcher agent that acts as a search tree node, spawning multiple children nodes with these same components and arbitrary depth and breadth.

3. Failure Modes of Existing Approaches

Despite recent progress, we still know little about how individual components in these systems perform, or fail. To study behavior on long-horizon tasks, we focus on BrowseComp, which naturally induces extended, multi-step search trajectories. For this task, the final outcome can reveal the overall performance of each framework as well as its relationship with the context window limitation and tool budget constraints. For this analysis, we let the framework run up to a fixed number of turns and output an answer. We categorize the final outcome in Table 2.

For this analysis, we consider different tool budgets for REACT and SEARCH-O1, and use the default 20 turns for HF-ODR. We observe that context window limitations and tool budgets are the main bottlenecks for existing approaches in Figure 3, and each framework exhibits distinct patterns.

Specifically, REACT often hits the context window limit over a long trajectory due to the large amount of text returned by each search call. As a result, it cannot effectively scale to long trajectories and make full use of its tool budgets. SEARCH-O1 failure cases are almost entirely due to exceeding the tool budget, which suggests increasing the tool budget may potentially lead to better performance. However, such an increase is non-trivial without incurring a significant amount of cost—each retrieval step in SEARCH-O1 involves scraping all search results, even though only a fraction of these results are relevant, leading to a large amount of LLM token consumption during the summarization step.

Finally, we observe that HF-ODR often prematurely terminates because the manager agent cannot leverage its search agent across multiple steps. Furthermore, HF-ODR is the only framework that do not use any tools in a significant percentage of the trajectories (10%), suggesting that complex prompt-engineered workflows may be prone to reducing the tool calling capabilities of the base model. The root cause of these failure modes is poor context management—exceeding context and tool budgets, or stopping too early. In the next section, we explore how to substantially improve agentic search through better context management.

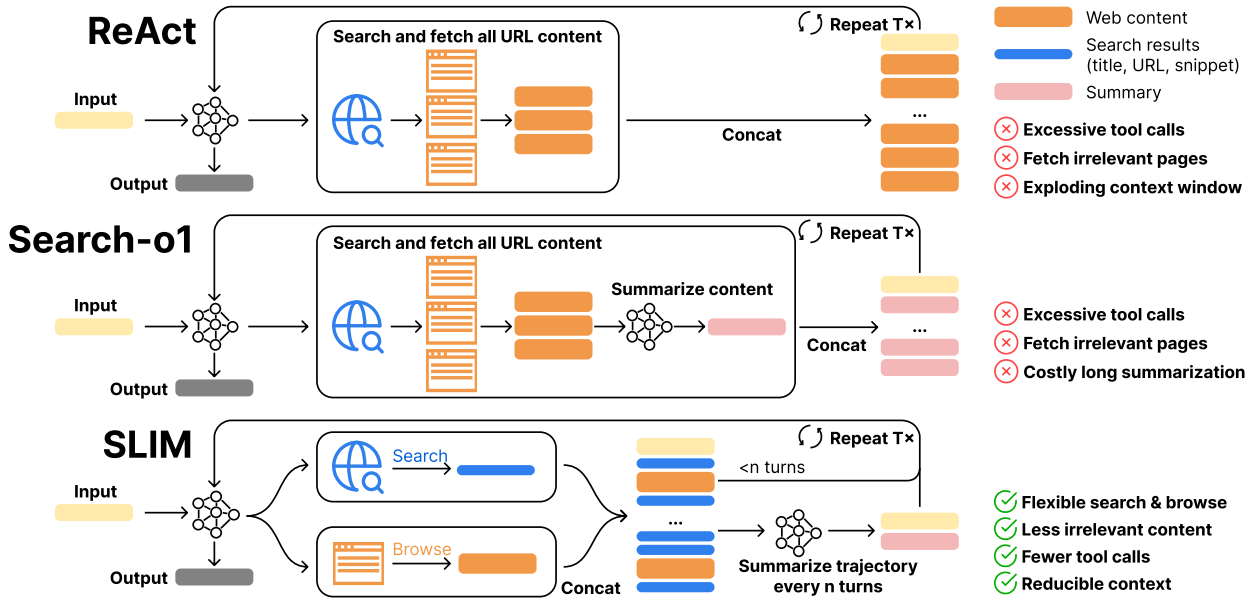

Figure 4. Compared to REACT and SEARCH-O1, the cooperation between search, browse, and summarization modules allows SLIM to accumulate shorter contexts and less noisy information after exploring the same amount of searches.

4. Our Framework: SLIM

A key takeaway from our analysis is that long-trajectory tasks require **scaling up the number of turns and tool calls while keeping the context concise to avoid hitting the context window limit**. Specifically, search results are often noisy and irrelevant to the answer, so filling up the context with content from all search results can lead to noisy context and unnecessary tool costs. Motivated by these observations, we introduce SLIM (**S**imple **L**ightweight **I**nformation **M**anagement) with two key principles: (1) using simple and flexible tools for LLMs to interact with, and (2) minimizing the amount of noisy information presented to the model and keeping the context concise during exploration. An overview of SLIM in comparison to existing frameworks is shown in Figure 4.

Concretely, SLIM adopts three simple yet powerful components—search, browse, and summarization—to effectively manage the context and scale the number of turns.

Search tool $\mathcal{R}$. SLIM uses a search tool that only returns the top k search results from a search engine, where each search result consists of a title, a URL, and a short snippet of its content. This is much more efficient than previous frameworks that return the full content for all search results, and relies on the main LLM to discern relevant context.

Browse tool $\mathcal{B}$. Our browse tool is designed to complement the search tool by allowing the LLM to dig deeper into promising search results. Specifically, the browse tool $\mathcal{B}(u, q) \rightarrow \max_{c_i \in c} \texttt{sim}(c_i, q)$ returns the most relevant section of the content c from the URL u to the query q, which enables the LLM to select the most relevant search result and choose a subset of the content that best matches the specific information it is looking for. As a result, our browse tool is significantly more efficient and cheaper than previous frameworks that exhaustively browse all search results in terms of both the scraping operations and the amount of new tokens introduced to the context.

Summarization module $\mathcal{S}$. Despite the brevity of each tool response, agent context inevitably grows as it explores over a long horizon of searches. To maintain a concise context while retaining the effective exploration history, we introduce a summarization module that periodically compresses the LLM context. We find a simple heuristic sufficient: we summarize the entire conversation history after every n turns of tool calls and replace the trajectory with the summary. This crucially differs from previous works where summarization is solely applied to search results at each turn.

Finally, we combine these components into a single framework by allowing the underlying LLM to call either the search or the browse tools at every turn. Then, the summarization module compresses the entire conversation every n turns to reduce the amount of noise. Our implementation uses Google[4] as the search tool, crawl4ai[5] as the browse tool, and the same LLM as the agent model for summarization. More details, an example trajectory, and ablations on

[4] https://serper.dev/
[5] https://github.com/unclecode/crawl4ai

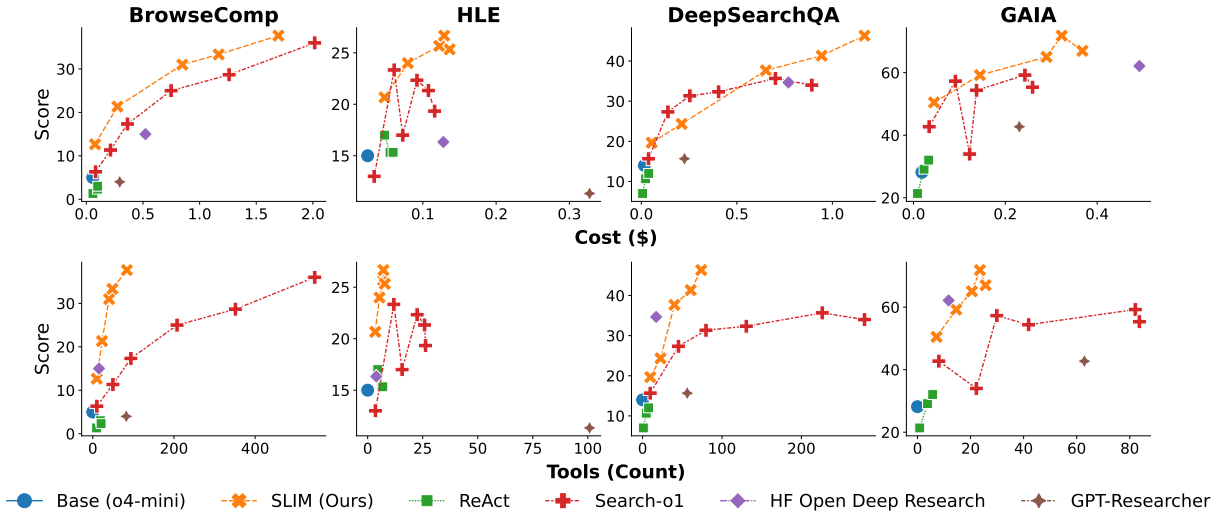

Figure 5. With o4-mini as the base model, SLIM consistently outperforms other baselines on BrowseComp while using fewer tool calls and lower overall costs. On HLE, SLIM can achieve overall higher performance and use fewer tool calls.

	Model	T	BrowseComp			HLE			DeepSearchQA			GAIA			HealthBench Hard		
			Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score
GLM-4.7	Base	-	2.7	0.9	2.0	1.8	0.8	9.0	3.1	0.7	8.0	2.6	0.8	18.4	1.5	0.3	12.9
	REACT	10	27.8	3.0	2.0	17.8	2.2	9.0	30.1	2.9	8.0	24.7	2.6	18.4	15.6	1.5	12.9
	SEARCH-O1	50	155.3	15.9	7.7	49.4	5.3	9.7	116.0	11.7	19.0	69.0	7.0	45.6	23.0	2.3	11.5
	SLIM	150	51.8	4.9	**17.3**	19.8	2.5	**13.7**	48.8	5.1	**20.7**	26.0	2.7	**49.5**	5.2	0.5	**13.7**
GPT-OSS	Base	-	0.0	0.2	2.7	0.0	0.2	10.0	0.0	0.1	9.3	0.0	0.1	23.3	0.0	0.1	36.6
	REACT	10	26.1	3.2	3.3	10.3	1.4	11.0	34.8	4.3	4.7	25.0	3.1	30.1	4.8	0.8	39.1
	SEARCH-O1	50	45.7	4.4	6.0	9.7	1.1	16.0	60.1	5.8	15.0	29.1	2.9	45.6	2.6	0.4	37.6
	SLIM	150	70.5	12.5	**20.0**	8.1	1.3	**19.0**	61.6	11.6	**34.3**	23.5	3.9	**58.3**	2.2	0.4	**39.3**

Table 3. SLIM performs the best across all settings. We evaluate open-weight models with settings of comparable cost. Costs are in US cents. More settings are available in §A.6.

		BrowseComp			HLE		
		Tools	Cost	Score	Tools	Cost	Score
Tongyi-DR-30B	-	0.0	2.8	2.3	0.0	2.2	11.0
REACT	10	39.0	3.5	1	28.0	3.0	12
SEARCH-O1	10	70.2	11.1	14.3	44.4	8.4	20.0
SLIM	150	61.6	8.3	19.7	23.1	5.3	19.7

Table 4. SLIM significantly outperforms SEARCH-O1 on BrowseComp and achieves similar performance on HLE with lower cost. Costs are in US cents.

the search tool, browse tool, and summarization module are shown in §A.4.

5. Results

We evaluate API models, o3, o4-mini, and Claude-4-Sonnet, open-weight models, GPT-OSS-120B (OpenAI et al., 2025) and GLM-4.7 Flash (Team et al., 2025), and RL-trained models, Tongyi-DeepResearch-30B (Team, 2025). For each instance, we evaluate the system's performance as well as the number of tool calls and tokens used. The number of tool calls is the sum of the search API and browse/scraping operations. Cost include the tool costs and the LLM token costs according to API prices, excluding cached tokens that are typically implemented in practical deep research systems. For each dataset we report results averaged over all instances. More details on the experimental setup can be found in §A.5.

We present the results with o3 and o4-mini as the base models in Figure 1 and 5. Under the same cost with o3, SLIM achieves significant improvements over SEARCH-O1, the best performing open-source framework, by 8, 6, 15, and 26 points on BrowseComp, HLE, DeepSearchQA, and GAIA, respectively. Similarly, with o4-mini, SLIM outperforms the open-source frameworks by 2, 4, 3, and 24 points on BrowseComp, HLE, DeepSearchQA, and GAIA, respectively. The detailed numbers with HealthBench Hard results and comparisons are shown in Table 14. We also conduct statistical tests to compare the performance of SLIM with the baselines, as shown in Table 16.

We also show results with different base models—GLM-4.7 Flash and GPT-OSS-120B in Table 3, Tongyi-DeepResearch-30B in Table 4, and Claude-4-Sonnet in Figure 8. SLIM consistently achieves the better or comparable performance at lower cost across these models and all

Question: Provide the birth name of a certain individual:
1. hired for a coordinator position in 2012 and later promoted
2. has a child that was born in the United States.
3. released their debut single between 2010 and 2015.

Groundtruth Answer: Nicholas Munene Mutum (a Kenyan actor)

Search queries:
1: debut single 2012 filipina actress model business administration
2: "child born in the United States" singer actress
3: "promoted to manager" "debut single" 2014
...
19: Filipina actress gave birth in the United States 2015
20: Filipina actress debut single 2013
...
48: "marketing coordinator" 2012 Philippines
49: "children born in the United States" actress "Philippines"

(1) Unfocused searches: overly generic queries that do not narrow down search space

(2) Confirmation bias: ≥50% search queries focus on a wrong candidates due to early noisy signals.

Search results:
1: wikipedia/Maja_Salvador, imdb/Filipina Beauty
2: timenote/Virginia_Weidler, wikipedia/Sharon_Pierre-Louis
...
20: timenote/Virginia_Weidler, wikipedia/Maja_Salvador
21: wikipedia/Nick_Mutuma

(3) Inefficient search: search repetitive information/URLs

(4) Answer ignored: correct answer found in trajectory but not used

Example Output 1:
Explanation: I was unable to reliably identify...
Exact Answer: Unable to determine from the information available

(5) Abstention: do not attempt to answer a question.

Example Output 2:
Explanation: Angeline Quinto satisfies every clue:
1. Angeline is a Filipino singer with a child born in the US.
2. Angeline's debut single was released in 2012.
3. Angeline was promoted from coordinator to manager at 1FMs

(6) Hallucination: generated statements are not supported by contents from the trajectory.
Cross check with search results → 2/4 unsupported statements

Figure 6. Examples of each trajectory-level failure mode on a BrowseComp sample.

datasets compared to other frameworks, suggesting that our simple design generalizes well to models of different sizes and training strategies. Notably, SLIM shows consistent trends across all three base models whereas certain frameworks only work well under certain models and datasets; for instance, HF-ODR performs well on DeepSearchQA and GAIA but poorly on BrowseComp, HealthBench, and HLE. Overall, this is strong evidence that SLIM serves as an effective framework for long-horizon tasks. Full results and additional ablations on different context management strategies (e.g., discarding context) and other design choices are in §A.6.

6. Fine-Grained Trajectory-level Analysis

6.1. Trajectory-Level Error Taxonomy

To understand how SLIM improves over other systems at a deeper level, we extend the analysis beyond the task outcome, and focus on characterizing the mistakes that a system makes over the course of its long search *trajectories*. To this end, we first develop a shared taxonomy of common failure modes by manually examining individual trajectories from the compared systems on BrowseComp. We present examples of each failure mode in the taxonomy in Figure 6, and detailed definitions in §A.3. Our taxonomy covers possible failure modes for long-horizon search agents in the information gathering process (e.g., unfocused searches, confirmation bias, and inefficient search) as well as the answer synthesis stage (e.g., ignoring the answer, abstention, and hallucination).

Based on the taxonomy, we develop an automated error analysis pipeline that annotates each trajectory with the failure modes using a mix of rule-based heuristics and LLM-as-a-judge approaches. Our pipeline carefully examines all parts of each trajectory—the search queries and results, the browsed contents, and the final answer—to identify the failure modes. We describe the pipeline more in §A.3.

6.2. Analysis of Trajectory-level Failure Modes

For fair comparison, we analyze all frameworks under a similar cost budget[6]: we choose the setting with the closest cost to SLIM with tool budget $T = 150$, according to Table 14. The distribution of trajectory-level errors are shown in Table 5, where we show the percentage of correct answer and each failure mode across all samples. We first observe

[6]We exclude GPT-R because their implementation do not return the contents of the search results.

Framework	Turn Budget	Correct	Confirm Bias	Unfocused Search	Inefficient Search	Abstention	Answer Ignored	Hallucinate
REACT	10	7.0	9.3	44.0	3.9	1.0	0.7	56.7
SEARCH-O1	50	48.3	9.3	33.7	7.2	4.3	26.0	46.8
HF-ODR	20	20.0	6.7	58.7	43.9	32.3	1.7	96.2
SLIM	150	56.0	9.7	34.0	7.6	27.7	30.7	19.0

Table 5. The percentage of trajectory over all samples that observe each failure mode. For hallucination only, we report the percentage of hallucinations for samples that ends with an incorrect answer and do not abstain.

that SLIM's advantage in performance could be attributed to the notably reduced hallucination rate compared to other frameworks. This is likely due to the fact that SLIM can choose what URLs to browse based on the search results, allowing it to reduce the amount of noise in the context. In contrast, the other frameworks observe significantly higher hallucination rates compared to SLIM, suggesting that they often resort to their parametric knowledge to answer the question when they cannot find the correct answer through tool calls.

Moreover, SEARCH-O1 and SLIM observe higher percentages of answer ignored than other frameworks. One explanation is that these frameworks tend to encounter more search results across their longer trajectories, which leads to a higher chance of finding the answer, but also a higher chance of ignoring it. In contrast, REACT and HF-ODR do not scale well to longer trajectories, which means they are unlikely to encounter the correct answer. Our analysis reveals that a promising direction for improving long-horizon agentic search is to enable language models to better identify the correct answer from long trajectories.

Notably, despite the improvements on hallucination, SLIM still suffers from high abstention rates, and is more prone to ignoring the groundtruth answers. We leave these improvements to future work, and hope that our trajectory-level analysis can be a useful tool for improving long-horizon systems in more interpretable and concrete ways.

7. Related Work

Long-horizon search agents. Recently, the community has taken great interests in search agents that extend beyond single-turn RAG by interleaving web search with reasoning across multiple turns (Yao et al., 2023). This paradigm has been actively developed in both industry (OpenAI, 2025; Google, 2025; xAI, 2025; Nguyen et al., 2025) and open-source communities (Jin et al., 2025b; Li et al., 2025b;a; Team, 2025, *inter alia*), advancing agent capabilities. However, as trajectories grow longer, these agents accumulate noisy context and become bounded by the context window, forcing early termination before tasks are fully resolved (Liu et al., 2024; Hong et al., 2025). To address this, recent efforts explore context management strategies that compress or prune trajectory history, preserving capacity for future steps. Concurrent to us, several works explore strategies such as *Summary* (Wu et al., 2025), *Discard-All* (Liu et al., 2025a), and *Keep-Recent-N* (Tang et al., 2025; Zeng et al., 2026), yet no clear winner has been observed. In contrast, we take a more holistic approach of long-horizon agentic search framework by carefully designing both the tool interfaces and context management. Notably, we also provide rigorous error analysis pipeline, enabling systematic comparison across different methodologies.

RL for long-horizon systems. Many recent works seek to improve search agents through reinforcement learning (Li et al., 2025c; Zheng et al., 2025; Chen et al., 2025; Li et al., 2025a; Wu et al., 2025, *inter alia*). A popular approach is to synthetically generate question-answer pairs that require long-horizon search trajectories (Xia et al., 2025; Tao et al., 2025). Other works focus on comparing different training objectives (Jin et al., 2025b;a). However, critical analysis of the error modes and comparison of different frameworks are still lacking.

8. Conclusion

In this work, we propose SLIM, a simple yet effective long-horizon agentic search framework that addresses context limitations prevalent in existing systems. We show that SLIM consistently achieves the highest performance across different proprietary and open models and diverse datasets compared to other frameworks while using fewer tool calls and lower overall costs, suggesting that our simple design enables better long-horizon agentic search. We also develop an automated error analysis pipeline to characterize the failure modes of long-horizon systems. Our analysis shows that SLIM is more resistant to failure modes such as hallucination. We hope our framework and analysis pipeline can serve as a useful tool for the community to understand and improve long-horizon agentic search systems.

References

Arora, R. K., Wei, J., Hicks, R. S., Bowman, P., Quiñonero-Candela, J., Tsimpourlas, F., Sharman, M., Shah, M.,

Vallone, A., Beutel, A., Heidecke, J., and Singhal, K. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

Belcak, P. and Molchanov, P. Universal deep research: Bring your own model and strategy, 2025. URL https://arxiv.org/abs/2509.00244.

Bohnet, B., Tran, V. Q., Verga, P., Aharoni, R., Andor, D., Soares, L. B., Eisenstein, J., Ganchev, K., Herzig, J., Hui, K., et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022. URL https://arxiv.org/pdf/2212.08037.pdf.

Chen, M., Sun, L., Li, T., Sun, H., Zhou, Y., Zhu, C., Wang, H., Pan, J. Z., Zhang, W., Chen, H., Yang, F., Zhou, Z., and Chen, W. Research: Learning to reason with search for llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2503.19470.

Elovic, A. gpt-researcher, July 2023. URL https://github.com/assafelovic/gpt-researcher.

Gangi Reddy, R., Mukherjee, S., Kim, J., Wang, Z., Hakkani-Tür, D., and Ji, H. Infogent: An agent-based framework for web information aggregation. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5745–5758, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.318. URL https://aclanthology.org/2025.findings-naacl.318/.

Gao, T., Yen, H., Yu, J., and Chen, D. Enabling large language models to generate text with citations. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Google. Gemini deep research — your personal research assistant, September 2025. URL https://gemini.google/overview/deep-research/.

Gupta, N., Chatterjee, R., Haas, L., Tao, C., Wang, A., Liu, C., Oiwa, H., Gribovskaya, E., Ackermann, J., Blitzer, J., Goldshtein, S., and Das, D. Deepsearchqa: Bridging the comprehensiveness gap for deep research agents, 2026. URL https://arxiv.org/abs/2601.20975.

Hong, K., Troynikov, A., and Huber, J. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma, July 2025. URL https://research.trychroma.com/context-rot.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24 (1), January 2023. ISSN 1532-4435.

Jin, B., Yoon, J., Kargupta, P., Arik, S. O., and Han, J. An empirical study on reinforcement learning for reasoning-search interleaved llm agents, 2025a. URL https://arxiv.org/abs/2505.15117.

Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S. O., Wang, D., Zamani, H., and Han, J. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=Rwhi91ideu.

Kamoi, R., Goyal, T., Rodriguez, J. D., and Durrett, G. WiCE: Real-World Entailment for Claims in Wikipedia. *arXiv preprint arXiv:2303.01432*, 2023. URL https://arxiv.org/abs/2303.01432.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Li, K., Zhang, Z., Yin, H., Zhang, L., Ou, L., Wu, J., Yin, W., Li, B., Tao, Z., Wang, X., Shen, W., Zhang, J., Zhang, D., Wu, X., Jiang, Y., Yan, M., Xie, P., Huang, F., and Zhou, J. Websailor: Navigating super-human reasoning for web agent, 2025a. URL https://arxiv.org/abs/2507.02592.

Li, X., Dong, G., Jin, J., Zhang, Y., Zhou, Y., Zhu, Y., Zhang, P., and Dou, Z. Search-o1: Agentic search-enhanced large reasoning models, 2025b. URL https://arxiv.org/abs/2501.05366.

Li, X., Jin, J., Dong, G., Qian, H., Zhu, Y., Wu, Y., Wen, J., and Dou, Z. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025c. doi: 10.48550/ARXIV.2504.21776. URL https://doi.org/10.48550/arXiv.2504.21776.

Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., et al. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Liu, J., Li, Y., Zhang, C., Li, J., Chen, A., Ji, K., Cheng, W., Wu, Z., Du, C., Xu, Q., Song, J., Zhu, Z., Chen, W.,

Zhao, P., and He, J. Webexplorer: Explore and evolve for training long-horizon web agents, 2025b. URL https://arxiv.org/abs/2509.06501.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., and Scialom, T. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

Nguyen, X.-P., Pandit, S., Reddy, R. G., Xu, A., Savarese, S., Xiong, C., and Joty, S. Sfr-deepresearch: Towards effective reinforcement learning for autonomously reasoning single agents, 2025. URL https://arxiv.org/abs/2509.06283.

OpenAI. Introducing deep research, February 2025. URL https://openai.com/index/introducing-deep-research/.

OpenAI, :, Agarwal, S., Ahmad, L., Ai, J., Altman, S., Applebaum, A., Arbus, E., Arora, R. K., Bai, Y., Baker, B., Bao, H., Barak, B., Bennett, A., Bertao, T., Brett, N., Brevdo, E., Brockman, G., Bubeck, S., Chang, C., Chen, K., Chen, M., Cheung, E., Clark, A., Cook, D., Dukhan, M., Dvorak, C., Fives, K., Fomenko, V., Garipov, T., Georgiev, K., Glaese, M., Gogineni, T., Goucher, A., Gross, L., Guzman, K. G., Hallman, J., Hehir, J., Heidecke, J., Helyar, A., Hu, H., Huet, R., Huh, J., Jain, S., Johnson, Z., Koch, C., Kofman, I., Kundel, D., Kwon, J., Kyrylov, V., Le, E. Y., Leclerc, G., Lennon, J. P., Lessans, S., Lezcano-Casado, M., Li, Y., Li, Z., Lin, J., Liss, J., Lily, Liu, Liu, J., Lu, K., Lu, C., Martinovic, Z., McCallum, L., McGrath, J., McKinney, S., McLaughlin, A., Mei, S., Mostovoy, S., Mu, T., Myles, G., Neitz, A., Nichol, A., Pachocki, J., Paino, A., Palmie, D., Pantuliano, A., Parascandolo, G., Park, J., Pathak, L., Paz, C., Peran, L., Pimenov, D., Pokrass, M., Proehl, E., Qiu, H., Raila, G., Raso, F., Ren, H., Richardson, K., Robinson, D., Rotsted, B., Salman, H., Sanjeev, S., Schwarzer, M., Sculley, D., Sikchi, H., Simon, K., Singhal, K., Song, Y., Stuckey, D., Sun, Z., Tillet, P., Toizer, S., Tsimpourlas, F., Vyas, N., Wallace, E., Wang, X., Wang, M., Watkins, O., Weil, K., Wendling, A., Whinnery, K., Whitney, C., Wong, H., Yang, L., Yang, Y., Yasunaga, M., Ying, K., Zaremba, W., Zhan, W., Zhang, C., Zhang, B., Zhang, E., and Zhao, S. gpt-oss-120b & gpt-oss-20b Model Card, 2025. URL https://arxiv.org/abs/2508.10925.

Phan, L., Gatti, A., Han, Z., Li, N., Hu, J., Zhang, H., Zhang, C. B. C., Shaaban, M., Ling, J., Shi, S., Choi, M., Agrawal, A., Chopra, A., Khoja, A., Kim, R., Ren, R., Hausenloy, J., Zhang, O., Mazeika, M., Dodonov, D., Nguyen, T., Lee, J., Anderson, D., Doroshenko, M., Stokes, A. C., Mahmood, M., Pokutnyi, O., Iskra, O., Wang, J. P., Levin, J.-C., Kazakov, M., Feng, F., Feng, S. Y., Zhao, H., Yu, M., Gangal, V., Zou, C., Wang, Z., Popov, S., Gerbicz, R., Galgon, G., Schmitt, J., Yeadon, W., Lee, Y., Sauers, S., Sanchez, A., Giska, F., Roth, M., Riis, S., Utpala, S., Burns, N., Goshu, G. M., Naiya, M. M., Agu, C., Giboney, Z., Cheatom, A., Fournier-Facio, F., Crowson, S.-J., Finke, L., Cheng, Z., Zampese, J., Hoerr, R. G., Nandor, M., Park, H., Gehrunger, T., Cai, J., McCarty, B., Garretson, A. C., Taylor, E., Sileo, D., Ren, Q., Qazi, U., Li, L., Nam, J., Wydallis, J. B., Arkhipov, P., Shi, J. W. L., Bacho, A., Willcocks, C. G., Cao, H., Motwani, S., de Oliveira Santos, E., Veith, J., Vendrow, E., Cojoc, D., Zenitani, K., Robinson, J., Tang, L., Li, Y., Vendrow, J., Fraga, N. W., Kuchkin, V., Maksimov, A. P., Marion, P., Efremov, D., Lynch, J., Liang, K., Mikov, A., Gritsevskiy, A., Guillod, J., Demir, G., Martinez, D., Pageler, B., Zhou, K., Soori, S., Press, O., Tang, H., Rissone, P., Green, S. R., Brüssel, L., Twayana, M., Dieuleveut, A., Imperial, J. M., Prabhu, A., Yang, J., Crispino, N., Rao, A., Zvonkine, D., Loiseau, G., Kalinin, M., Lukas, M., Manolescu, C., Stambaugh, N., Mishra, S., Hogg, T., Bosio, C., Coppola, B. P., Salazar, J., Jin, J., Sayous, R., Ivanov, S., Schwaller, P., Senthilkuma, S., Bran, A. M., Algaba, A., den Houte, K. V., Sypt, L. V. D., Verbeken, B., Noever, D., Kopylov, A., Myklebust, B., Li, B., Schut, L., Zheltonozhskii, E., Yuan, Q., Lim, D., Stanley, R., Yang, T., Maar, J., Wykowski, J., Oller, M., Sahu, A., Ardito, C. G., Hu, Y., Kamdoum, A. G. K., Jin, A., Vilchis, T. G., Zu, Y., Lackner, M., Koppel, J., Sun, G., Antonenko, D. S., Chern, S., Zhao, B., Arsene, P., Cavanagh, J. M., Li, D., Shen, J., Crisostomi, D., Zhang, W., Dehghan, A., Ivanov, S., Perrella, D., Kaparov, N., Zang, A., Sucholutsky, I., Kharlamova, A., Orel, D., Poritski, V., Ben-David, S., Berger, Z., Whitfill, P., Foster, M., Munro, D., Ho, L., Sivarajan, S., Hava, D. B., Kuchkin, A., Holmes, D., Rodriguez-Romero, A., Sommerhage, F., Zhang, A., Moat, R., Schneider, K., Kazibwe, Z., Clarke, D., Kim, D. H., Dias, F. M., Fish, S., Elser, V., Kreiman, T., Vilchis, V. E. G., Klose, I., Anantheswaran, U., Zweiger, A., Rawal, K., Li, J., Nguyen, J., Daans, N., Heidinger, H., Radionov, M., Rozhoň, V., Ginis, V., Stump, C., Cohen, N., Poświata, R., Tkadlec, J., Goldfarb, A., Wang, C., Padlewski, P., Barzowski, S., Montgomery, K., Stendall, R., Tucker-Foltz, J., Stade, J., Rogers, T. R., Goertzen, T., Grabb, D., Shukla, A., Givré, A., Ambay, J. A., Sen, A., Aziz, M. F., Inlow, M. H., He, H., Zhang, L., Kaddar, Y., Ängquist, I., Chen, Y., Wang, H. K., Ramakrishnan, K., Thornley, E., Terpin, A., Schoelkopf, H., Zheng, E., Carmi, A., Brown, E. D. L., Zhu, K., Bartolo, M., Wheeler, R., Stehberger, M., Brad-

shaw, P., Heimonen, J., Sridhar, K., Akov, I., Sandlin, J., Makarychev, Y., Tam, J., Hoang, H., Cunningham, D. M., Goryachev, V., Patramanis, D., Krause, M., Redenti, A., Aldous, D., Lai, J., Coleman, S., Xu, J., Lee, S., Magoulas, I., Zhao, S., Tang, N., Cohen, M. K., Paradise, O., Kirchner, J. H., Ovchynnikov, M., Matos, J. O., Shenoy, A., Wang, M., Nie, Y., Sztyber-Betley, A., Faraboschi, P., Riblet, R., Crozier, J., Halasyamani, S., Verma, S., Joshi, P., Meril, E., Ma, Z., Andréoletti, J., Singhal, R., Platnick, J., Nevirkovets, V., Basler, L., Ivanov, A., Khoury, S., Gustafsson, N., Piccardo, M., Mostaghimi, H., Chen, Q., Singh, V., Khánh, T. Q., Rosu, P., Szlyk, H., Brown, Z., Narayan, H., Menezes, A., Roberts, J., Alley, W., Sun, K., Patel, A., Lamparth, M., Reuel, A., Xin, L., Xu, H., Loader, J., Martin, F., Wang, Z., Achilleos, A., Preu, T., Korbak, T., Bosio, I., Kazemi, F., Chen, Z., Bálint, B., Lo, E. J. Y., Wang, J., Nunes, M. I. S., Milbauer, J., Bari, M. S., Wang, Z., Ansarinejad, B., Sun, Y., Durand, S., Elgnainy, H., Douville, G., Tordera, D., Balabanian, G., Wolff, H., Kvistad, L., Milliron, H., Sakor, A., Eron, M., O., A. F. D., Shah, S., Zhou, X., Kamalov, F., Abdoli, S., Santens, T., Barkan, S., Tee, A., Zhang, R., Tomasiello, A., Luca, G. B. D., Looi, S.-Z., Le, V.-K., Kolt, N., Pan, J., Rodman, E., Drori, J., Fossum, C. J., Muennighoff, N., Jagota, M., Pradeep, R., Fan, H., Eicher, J., Chen, M., Thaman, K., Merrill, W., Firsching, M., Harris, C., Ciobâcă, S., Gross, J., Pandey, R., Gusev, I., Jones, A., Agnihotri, S., Zhelnov, P., Mofayezi, M., Piperski, A., Zhang, D. K., Dobarskyi, K., Leventov, R., Soroko, I., Duersch, J., Taamazyan, V., Ho, A., Ma, W., Held, W., Xian, R., Zebaze, A. R., Mohamed, M., Leser, J. N., Yuan, M. X., Yacar, L., Lengler, J., Olszewska, K., Fratta, C. D., Oliveira, E., Jackson, J. W., Zou, A., Chidambaram, M., Manik, T., Haffenden, H., Stander, D., Dasouqi, A., Shen, A., Golshani, B., Stap, D., Kretov, E., Uzhou, M., Zhidkovskaya, A. B., Winter, N., Rodriguez, M. O., Lauff, R., Wehr, D., Tang, C., Hossain, Z., Phillips, S., Samuele, F., Ekström, F., Hammon, A., Patel, O., Farhidi, F., Medley, G., Mohammadzadeh, F., Peñaflor, M., Kassahun, H., Friedrich, A., Perez, R. H., Pyda, D., Sakal, T., Dhamane, O., Mirabadi, A. K., Hallman, E., Okutsu, K., Battaglia, M., Maghsoudimehrabani, M., Amit, A., Hulbert, D., Pereira, R., Weber, S., Handoko, Peristyy, A., Malina, S., Mehkary, M., Aly, R., Reidegeld, F., Dick, A.-K., Friday, C., Singh, M., Shapourian, H., Kim, W., Costa, M., Gurdogan, H., Kumar, H., Ceconello, C., Zhuang, C., Park, H., Carroll, M., Tawfeek, A. R., Steinerberger, S., Aggarwal, D., Kirchhof, M., Dai, L., Kim, E., Ferret, J., Shah, J., Wang, Y., Yan, M., Burdzy, K., Zhang, L., Franca, A., Pham, D. T., Loh, K. Y., Robinson, J., Jackson, A., Giordano, P., Petersen, P., Cosma, A., Colino, J., White, C., Votava, J., Vinnikov, V., Delaney, E., Spelda, P., Stritecky, V., Shahid, S. M., Mourrat, J.-C., Vetoshkin, L., Sponselee, K., Bacho, R., Yong, Z.-X., de la Rosa, F., Cho, N., Li, X., Malod, G., Weller, O., Albani, G., Lang, L., Laurendeau, J., Kazakov, D., Adesanya, F., Portier, J., Hollom, L., Souza, V., Zhou, Y. A., Degorre, J., Yalın, Y., Obikoya, G. D., Rai, Bigi, F., Boscá, M. C., Shumar, O., Bacho, K., Recchia, G., Popescu, M., Shulga, N., Tanwie, N. M., Lux, T. C. H., Rank, B., Ni, C., Brooks, M., Yakimchyk, A., Huanxu, Liu, Cavalleri, S., Häggström, O., Verkama, E., Newbould, J., Gundlach, H., Brito-Santana, L., Amaro, B., Vajipey, V., Grover, R., Wang, T., Kratish, Y., Li, W.-D., Gopi, S., Caciolai, A., de Witt, C. S., Hernández-Cámara, P., Rodolà, E., Robins, J., Williamson, D., Cheng, V., Raynor, B., Qi, H., Segev, B., Fan, J., Martinson, S., Wang, E. Y., Hausknecht, K., Brenner, M. P., Mao, M., Demian, C., Kassani, P., Zhang, X., Avagian, D., Scipio, E. J., Ragoler, A., Tan, J., Sims, B., Plecnik, R., Kirtland, A., Bodur, O. F., Shinde, D. P., Labrador, Y. C. L., Adoul, Z., Zekry, M., Karakoc, A., Santos, T. C. B., Shamseldeen, S., Karim, L., Liakhovitskaia, A., Resman, N., Farina, N., Gonzalez, J. C., Maayan, G., Anderson, E., Pena, R. D. O., Kelley, E., Mariji, H., Pouriamanesh, R., Wu, W., Finocchio, R., Alarab, I., Cole, J., Ferreira, D., Johnson, B., Safdari, M., Dai, L., Arthornthurasuk, S., McAlister, I. C., Moyano, A. J., Pronin, A., Fan, J., Ramirez-Trinidad, A., Malysheva, Y., Pottmaier, D., Taheri, O., Stepanic, S., Perry, S., Askew, L., Rodríguez, R. A. H., Minissi, A. M. R., Lorena, R., Iyer, K., Fasiludeen, A. A., Clark, R., Ducey, J., Piza, M., Somrak, M., Vergo, E., Qin, J., Borbás, B., Chu, E., Lindsey, J., Jallon, A., McInnis, I. M. J., Chen, E., Semler, A., Gloor, L., Shah, T., Carauleanu, M., Lauer, P., Huy, T. D., Shahrtash, H., Duc, E., Lewark, L., Brown, A., Albanie, S., Weber, B., Vaz, W. S., Clavier, P., Fan, Y., e Silva, G. P. R., Long, Lian, Abramovitch, M., Jiang, X., Mendoza, S., Islam, M., Gonzalez, J., Mavroudis, V., Xu, J., Kumar, P., Goswami, L. P., Bugas, D., Heydari, N., Jeanplong, F., Jansen, T., Pinto, A., Apronti, A., Galal, A., Ze-An, N., Singh, A., Jiang, T., of Arc Xavier, J., Agarwal, K. P., Berkani, M., Zhang, G., Du, Z., de Oliveira Junior, B. A., Malishev, D., Remy, N., Hartman, T. D., Tarver, T., Mensah, S., Loume, G. A., Morak, W., Habibi, F., Hoback, S., Cai, W., Gimenez, J., Montecillo, R. G., Łucki, J., Campbell, R., Sharma, A., Meer, K., Gul, S., Gonzalez, D. E., Alapont, X., Hoover, A., Chhablani, G., Vargus, F., Agarwal, A., Jiang, Y., Patil, D., Outevsky, D., Scaria, K. J., Maheshwari, R., Dendane, A., Shukla, P., Cartwright, A., Bogdanov, S., Mündler, N., Möller, S., Arnaboldi, L., Thaman, K., Siddiqi, M. R., Saxena, P., Gupta, H., Fruhauff, T., Sherman, G., Vincze, M., Usawasutsakorn, S., Ler, D., Radhakrishnan, A., Enyekwe, I., Salauddin, S. M., Muzhen, J., Maksapetyan, A., Rossbach, V., Harjadi, C., Bahaloohoreh, M., Sparrow, C., Sidhu, J., Ali, S., Bian, S., Lai, J., Singer, E., Uro, J. L., Bateman, G., Sayed, M., Menshawy, A., Duclosel, D., Bezzi, D., Jain, Y., Aaron, A., Tiryakioglu, M., Siddh, S.,

Krenek, K., Shah, I. A., Jin, J., Creighton, S., Peskoff, D., EL-Wasif, Z., V, R. P., Richmond, M., McGowan, J., Patwardhan, T., Sun, H.-Y., Sun, T., Zubić, N., Sala, S., Ebert, S., Kaddour, J., Schottdorf, M., Wang, D., Petruzella, G., Meiburg, A., Medved, T., ElSheikh, A., Hebbar, S. A., Vaquero, L., Yang, X., Poulos, J., Zouhar, V., Bogdanik, S., Zhang, M., Sanz-Ros, J., Anugraha, D., Dai, Y., Nhu, A. N., Wang, X., Demircali, A. A., Jia, Z., Zhou, Y., Wu, J., He, M., Chandok, N., Sinha, A., Luo, G., Le, L., Noyé, M., Perełkiewicz, M., Pantidis, I., Qi, T., Purohit, S. S., Parcalabescu, L., Nguyen, T.-H., Winata, G. I., Ponti, E. M., Li, H., Dhole, K., Park, J., Abbondanza, D., Wang, Y., Nayak, A., Caetano, D. M., Wong, A. A. W. L., del Rio-Chanona, M., Kondor, D., Francois, P., Chalstrey, E., Zsambok, J., Hoyer, D., Reddish, J., Hauser, J., Rodrigo-Ginés, F.-J., Datta, S., Shepherd, M., Kamphuis, T., Zhang, Q., Kim, H., Sun, R., Yao, J., Dernoncourt, F., Krishna, S., Rismanchian, S., Pu, B., Pinto, F., Wang, Y., Shridhar, K., Overholt, K. J., Briia, G., Nguyen, H., David, Bartomeu, S., Pang, T. C., Wecker, A., Xiong, Y., Li, F., Huber, L. S., Jaeger, J., Maddalena, R. D., Lù, X. H., Zhang, Y., Beger, C., Kon, P. T. J., Li, S., Sanker, V., Yin, M., Liang, Y., Zhang, X., Agrawal, A., Yifei, L. S., Zhang, Z., Cai, M., Sonmez, Y., Cozianu, C., Li, C., Slen, A., Yu, S., Park, H. K., Sarti, G., Briański, M., Stolfo, A., Nguyen, T. A., Zhang, M., Perlitz, Y., Hernandez-Orallo, J., Li, R., Shabani, A., Juefei-Xu, F., Dhingra, S., Zohar, O., Nguyen, M. C., Pondaven, A., Yilmaz, A., Zhao, X., Jin, C., Jiang, M., Todoran, S., Han, X., Kreuer, J., Rabern, B., Plassart, A., Maggetti, M., Yap, L., Geirhos, R., Kean, J., Wang, D., Mollaei, S., Sun, C., Yin, Y., Wang, S., Li, R., Chang, Y., Wei, A., Bizeul, A., Wang, X., Arrais, A. O., Mukherjee, K., Chamorro-Padial, J., Liu, J., Qu, X., Guan, J., Bouyamourn, A., Wu, S., Plomecka, M., Chen, J., Tang, M., Deng, J., Subramanian, S., Xi, H., Chen, H., Zhang, W., Ren, Y., Tu, H., Kim, S., Chen, Y., Marjanović, S. V., Ha, J., Luczyna, G., Ma, J. J., Shen, Z., Song, D., Zhang, C. E., Wang, Z., Gendron, G., Xiao, Y., Smucker, L., Weng, E., Lee, K. H., Ye, Z., Ermon, S., Lopez-Miguel, I. D., Knights, T., Gitter, A., Park, N., Wei, B., Chen, H., Pai, K., Elkhanany, A., Lin, H., Siedler, P. D., Fang, J., Mishra, R., Zsolnai-Fehér, K., Jiang, X., Khan, S., Yuan, J., Jain, R. K., Lin, X., Peterson, M., Wang, Z., Malusare, A., Tang, M., Gupta, I., Fosin, I., Kang, T., Dworakowska, B., Matsumoto, K., Zheng, G., Sewuster, G., Villanueva, J. P., Rannev, I., Chernyavsky, I., Chen, J., Banik, D., Racz, B., Dong, W., Wang, J., Bashmal, L., Gonçalves, D. V., Hu, W., Bar, K., Bohdal, O., Patlan, A. S., Dhuliawala, S., Geirhos, C., Wist, J., Kansal, Y., Chen, B., Tire, K., Yücel, A. T., Christof, B., Singla, V., Song, Z., Chen, S., Ge, J., Ponkshe, K., Park, I., Shi, T., Ma, M. Q., Mak, J., Lai, S., Moulin, A., Cheng, Z., Zhu, Z., Zhang, Z., Patil, V., Jha, K., Men, Q., Wu, J., Zhang, T., Vieira, B. H., Aji, A. F., Chung, J.-W., Mahfoud, M., Hoang, H. T., Sperzel, M., Hao, W., Meding, K., Xu, S., Kostakos, V., Manini, D., Liu, Y., Toukmaji, C., Paek, J., Yu, E., Demircali, A. E., Sun, Z., Dewerpe, I., Qin, H., Pflugfelder, R., Bailey, J., Morris, J., Heilala, V., Rosset, S., Yu, Z., Chen, P. E., Yeo, W., Jain, E., Yang, R., Chigurupati, S., Chernyavsky, J., Reddy, S. P., Venugopalan, S., Batra, H., Park, C. F., Tran, H., Maximiano, G., Zhang, G., Liang, Y., Shiyu, H., Xu, R., Pan, R., Suresh, S., Liu, Z., Gulati, S., Zhang, S., Turchin, P., Bartlett, C. W., Scotese, C. R., Cao, P. M., Nattanmai, A., McKellips, G., Cheraku, A., Suhail, A., Luo, E., Deng, M., Luo, J., Zhang, A., Jindel, K., Paek, J., Halevy, K., Baranov, A., Liu, M., Avadhanam, A., Zhang, D., Cheng, V., Ma, B., Fu, E., Do, L., Lass, J., Yang, H., Sunkari, S., Bharath, V., Ai, V., Leung, J., Agrawal, R., Zhou, A., Chen, K., Kalpathi, T., Xu, Z., Wang, G., Xiao, T., Maung, E., Lee, S., Yang, R., Yue, R., Zhao, B., Yoon, J., Sun, S., Singh, A., Luo, E., Peng, C., Osbey, T., Wang, T., Echeazu, D., Yang, H., Wu, T., Patel, S., Kulkarni, V., Sundarapandiyan, V., Zhang, A., Le, A., Nasim, Z., Yalam, S., Kasamsetty, R., Samal, S., Yang, H., Sun, D., Shah, N., Saha, A., Zhang, A., Nguyen, L., Nagumalli, L., Wang, K., Zhou, A., Wu, A., Luo, J., Telluri, A., Yue, S., Wang, A., and Hendrycks, D. Humanity's last exam, 2025. URL https://arxiv.org/abs/2501.14249.

Qiao, Z., Chen, G., Chen, X., Yu, D., Yin, W., Wang, X., Zhang, Z., Li, B., Yin, H., Li, K., Min, R., Liao, M., Jiang, Y., Xie, P., Huang, F., and Zhou, J. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents, 2025. URL https://arxiv.org/abs/2509.13309.

Rashkin, H., Nikolaev, V., Lamm, M., Aroyo, L., Collins, M., Das, D., Petrov, S., Tomar, G. S., Turc, I., and Reitter, D. Measuring Attribution in Natural Language Generation Models. *Computational Linguistics*, pp. 1–64, 08 2023. ISSN 0891-2017. doi: 10.1162/coli_a_00486. URL https://doi.org/10.1162/coli_a_00486.

Robertson, S. and Zaragoza, H. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, apr 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Roucher, A., del Moral, A. V., Noyan, M., Wolf, T., and Fourrier, C. Open-source DeepResearch – Freeing our search agents, January 2025. URL https://huggingface.co/blog/open-deep-research.

Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., and Yih, W.-t. REPLUG: Retrieval-augmented black-box language models. In Duh, K.,

Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463. URL https://aclanthology.org/2024.naacl-long.463/.

Tang, Q., Xiang, H., Yu, L., Yu, B., Lu, Y., Han, X., Sun, L., Zhang, W., Wang, P., Liu, S., et al. Beyond turn limits: Training deep search agents with dynamic context window. *arXiv preprint arXiv:2510.08276*, 2025.

Tao, Z., Wu, J., Yin, W., Zhang, J., Li, B., Shen, H., Li, K., Zhang, L., Wang, X., Jiang, Y., Xie, P., Huang, F., and Zhou, J. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

Team, G., Zeng, A., Lv, X., Zheng, Q., Hou, Z., Chen, B., Xie, C., Wang, C., Yin, D., Zeng, H., Zhang, J., Wang, K., Zhong, L., Liu, M., Lu, R., Cao, S., Zhang, X., Huang, X., Wei, Y., Cheng, Y., An, Y., Niu, Y., Wen, Y., Bai, Y., Du, Z., Wang, Z., Zhu, Z., Zhang, B., Wen, B., Wu, B., Xu, B., Huang, C., Zhao, C., Cai, C., Yu, C., Li, C., Ge, C., Huang, C., Zhang, C., Xu, C., Zhu, C., Li, C., Yin, C., Lin, D., Yang, D., Jiang, D., Ai, D., Zhu, E., Wang, F., Pan, G., Wang, G., Sun, H., Li, H., Li, H., Hu, H., Zhang, H., Peng, H., Tai, H., Zhang, H., Wang, H., Yang, H., Liu, H., Zhao, H., Liu, H., Yan, H., Liu, H., Chen, H., Li, J., Zhao, J., Ren, J., Jiao, J., Zhao, J., Yan, J., Wang, J., Gui, J., Zhao, J., Liu, J., Li, J., Li, J., Lu, J., Wang, J., Yuan, J., Li, J., Du, J., Du, J., Liu, J., Zhi, J., Gao, J., Wang, K., Yang, L., Xu, L., Fan, L., Wu, L., Ding, L., Wang, L., Zhang, M., Li, M., Xu, M., Zhao, M., Zhai, M., Du, P., Dong, Q., Lei, S., Tu, S., Yang, S., Lu, S., Li, S., Li, S., Shuang-Li, Yang, S., Yi, S., Yu, T., Tian, W., Wang, W., Yu, W., Tam, W. L., Liang, W., Liu, W., Wang, X., Jia, X., Gu, X., Ling, X., Wang, X., Fan, X., Pan, X., Zhang, X., Zhang, X., Fu, X., Zhang, X., Xu, Y., Wu, Y., Lu, Y., Wang, Y., Zhou, Y., Pan, Y., Zhang, Y., Wang, Y., Li, Y., Su, Y., Geng, Y., Zhu, Y., Yang, Y., Li, Y., Wu, Y., Li, Y., Liu, Y., Wang, Y., Li, Y., Zhang, Y., Liu, Z., Yang, Z., Zhou, Z., Qiao, Z., Feng, Z., Liu, Z., Zhang, Z., Wang, Z., Yao, Z., Wang, Z., Liu, Z., Chai, Z., Li, Z., Zhao, Z., Chen, W., Zhai, J., Xu, B., Huang, M., Wang, H., Li, J., Dong, Y., and Tang, J. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Team, T. D. Tongyi deepresearch: A new era of open-source ai researchers. https://github.com/Alibaba-NLP/DeepResearch, 2025.

Wang, X., Chen, Y., Yuan, L., Zhang, Y., Li, Y., Peng, H., and Ji, H. Executable code actions elicit better llm agents. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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, 2025. URL https://arxiv.org/abs/2504.12516.

Wu, X., Li, K., Zhao, Y., Zhang, L., Ou, L., Yin, H., Zhang, Z., Jiang, Y., Xie, P., Huang, F., Cheng, M., Wang, S., Cheng, H., and Zhou, J. Resum: Unlocking long-horizon search intelligence via context summarization, 2025. URL https://arxiv.org/abs/2509.13313.

xAI. Grok 3 beta — the age of reasoning agents, February 2025. URL https://x.ai/news/grok-3.

Xia, Z., Luo, K., Qian, H., and Liu, Z. Open data synthesis for deep research, 2025. URL https://arxiv.org/abs/2509.00375.

Xu, R. and Peng, J. A comprehensive survey of deep research: Systems, methodologies, and applications, 2025. URL https://arxiv.org/abs/2506.12594.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

Yen, H., Gao, T., Hou, M., Ding, K., Fleischer, D., Izsak, P., Wasserblat, M., and Chen, D. Helmet: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations (ICLR)*, 2025.

Zeng, A., Lv, X., Hou, Z., Du, Z., Zheng, Q., Chen, B., Yin, D., Ge, C., Xie, C., Wang, C., et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Zheng, Y., Fu, D., Hu, X., Cai, X., Ye, L., Lu, P., and Liu, P. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments, 2025. URL https://arxiv.org/abs/2504.03160.

A. Appendix

A.1. Existing Frameworks

REACT (Yao et al., 2023) is a simple framework that allows an LLM agent to alternate between thinking and acting. This framework allows the agent to use tool calls across many turns. Following the original work's knowledge-intensive

task settings, our implementation gives the LLM access to a single search tool—given a query, the tool returns a list of top 10 search results, from a search engine, along with their web contents. The search results are then concatenated and appended to the agent context for subsequent steps. When the LLM chooses not to use the search tool, the final output is used for evaluation.

In our implementation, we vary the maximum number of turns in each trajectory from 1 to 10. Consistent with SLIM, we use Google as the search engine, accessed through the Serper API, which returns a list of top 10 search results. Each search result contains a title, a URL, and a short snippet of the content. After obtaining the top 10 search results, we emulate previous RAG approaches by scraping all search result URLs and concatenate their content. Similar to SLIM, we use crawl4ai to scrape web pages. We truncate each scraped document to at most 10,000 characters, which corresponds to roughly 1,000 tokens.

We notice that REACT often hits the context window limit as the retrieval results are often too long. When the LLM API call fails due to the context window limit, we fallback to not using any tools and just ask the base LLM to answer the question. As a result, we only experiment with up to 10 turns, where the framework already falls back to not using any tools for most queries. A sketch of the framework is shown in Alg. 1.

SEARCH-O1 (Li et al., 2025b) builds upon REACT with an additional "reason-in-document" step, where an LLM summarizes the list of top 10 search results and their contents before appending the results to the agent's input context. Although the summary added to the agent context is relatively short compared to the full search result, this approach still uses a large amount of browse calls in each search step, and the summarization steps incur a large amount of LLM token usage. In our setting, we vary the maximum number of turns in each trajectory from 1 to up to 100 turns.

Similar to REACT, the retrieval tool at each step consists of a single Serper API call, followed by multiple scraping operations. We adopt the code from the original implementation[7], which uses BeautifulSoup[8] to scrape the search result URLs. In this implementation, the scraping operation will extract part of the web content that best matches the short snippet returned by the search engine. The matching is done by simply computing the F1 scores between the snippet and sentences in the web page. Subsequently, the context is filled up with at most 2,500 characters from the web page. Then, all context from the search results are concatenated and appended to the agent context for the summarization step.

It is important to note that the scraping operation is relatively expensive due to the network latency, resulting in long running time for the framework. A sketch of the framework is shown in Alg. 2.

HuggingFace OpenDeepResearch (HF-ODR; Roucher et al., 2025) leverages a hierarchical structure consisting of a manager agent and a search agent. The manager agent calls the search agent to perform detailed searches, and the search agent iteratively interacts with the search engine and a simulated browser to gather information. When the search agent concludes its searches, it generates a summary of its searches and returns it to the manager agent. The manager agent may use the summary to issue additional queries or output the final answer. Furthermore, another key feature of HF-ODR is its access to additional tools, such as a Python interpreter. We use the default settings[9], which fixes the maximum number of turns for the manager and search agent to be 20. A sketch of the framework is shown in Alg. 3. Specific descriptions of each tool can be found in Section A.2.

GPT-Researcher (GPT-R; Elovic, 2023) is a complex multi-agent system where each agent has distinct roles. Specifically, the system consists of a researcher conductor that orchestrates the search process, a report generator that generates the final report at the end of the search process, a context manager that summarizes search results, and a source curator that selects relevant sources from scraped web pages. Finally, GPT-R uses a deep researcher agent that acts as the node of a search tree, where each node is able to spawn multiple child nodes, each of which is a system with the previously described components. We use the default settings of the framework[10], which fixes the depth of the search tree to be 2 and the breadth of search at each depth to be 4. A sketch of the framework is shown in Alg. 4.

Other frameworks. There are many recent works on agentic search systems and memory-management frameworks (Gangi Reddy et al., 2025; Xu & Peng, 2025; Belcak & Molchanov, 2025). We chose the most popular open-source agentic search and deep research systems for comparison. These systems also span both simple single-agent and complex multi-agent systems, which we believe serve as a representative and fair group of baselines for the paper. Due to the high cost and long runtime of agentic systems, we

[7] https://github.com/RUC-NLPIR/Search-o1

[8] https://beautiful-soup-4.readthedocs.io/en/latest/

[9] https://github.com/huggingface/smolagents/tree/main/examples/open_deep_research

[10] https://github.com/assafelovic/gpt-researcher

only evaluate the representative baselines. Although there are explicit memory-management frameworks, we find that existing summarization models already do something similar to memory-selective mechanisms through qualitative analysis. In the example trajectory we show in Figure 7, the model summarizes the trajectory into several bullet points, such as "Investigation and findings so far", "Current hypothesis", and "Needed next". The resulting summary is similar to many memory-selective mechanisms that only retain relevant facts to the current query. Thus, we find that allowing the model to compress the full trajectory naturally filters out irrelevant information while achieving simplicity and avoiding over-prompt-engineering.

Algorithm 1: ReAct

Data: Task input x, LLM θ, maximum number of turns T
Function *search(q)*:
  return $(title_i, url_i, snippet_i)_{i=1}^{k}$;
Function *browse(u, q)*:
  $D \leftarrow \text{scrape}(u)$;
  return $D[:10000]$;
Result: Task output y
Turn $t \leftarrow 1$;
Context $C \leftarrow \{x\}$;
$\mathcal{T} \leftarrow \{\text{search}\}$;
while $t < T$ **do**
  $o_t \leftarrow \theta(C; \mathcal{T})$; /* LLM may only call the search tool */
  switch o_t **do**
    case *search* **do**
      $R \leftarrow \text{search}(o_t)$; /* Perform search */
      $C \leftarrow C \cup \{o_t\}$; /* Browse every search result and append */
      for $(t_i, u_i, s_i) \in R$ **do**
        $C \leftarrow C \cup \text{browse}(u_i, s_i)$
    case *Final Answer* **do**
      return o_t;
  $t \leftarrow t + 1$;
return $\theta(C; \textit{final answer})$;

A.2. HuggingFace Open Deep Research Tools

HF-ODR is a hierarchical framework that consists of a manager agent and a search agent. The manager agent has access to the following tools:

1. **Search Agent**: an agent that will search the internet to answer a question.
2. **Visualizer**: given the path to a downloaded image, it will call an LLM to answer questions about the image.
3. **Text Inspector**: given the path to a downloaded text file, it will call an LLM to answer questions about the text.

Algorithm 2: Search-o1

Data: Task input x, LLM θ, maximum number of turns T, summary interval n
Function *search(q)*:
  return $(title_i, url_i, snippet_i)_{i=1}^{k}$;
Function *visit(u, q)*:
  $D \leftarrow \text{scrape}(u)$;
  $D \leftarrow \text{split}(D) = \{d_i\}_{i=1}^{m}$;
  if $q = \emptyset$ then **return** $d' \leftarrow d_1$;
  else $d' \leftarrow \arg\max_{d_i \in D} \text{F1}(d_i, q)$;
  return d';
Result: Task output y
Turn $t \leftarrow 1$;
Context $C \leftarrow \{x\}$;
$\mathcal{T} \leftarrow \{\text{search}\}$;
while $t < T$ **do**
  $o_t \leftarrow \theta(C; \mathcal{T})$; /* LLM may only call the search tool */
  switch o_t **do**
    case *search* **do**
      $R \leftarrow \text{search}(o_t)$; /* Perform search */
      $l \leftarrow \text{length}(C)$;
      $D \leftarrow \{c_i\}_{i=l-5}^{l}$;
      for $(t_i, u_i, s_i) \in R$ **do**
        $D \leftarrow D \cup \text{visit}(u_i, s_i)$; /* Visit every search result */
      $C \leftarrow C \cup \{o_t, \theta(D; \text{summarize})\}$;
    case *Final Answer* **do**
      return o_t;
  $t \leftarrow t + 1$;
return $\theta(C; \textit{final answer})$;

The search agent has access to the following tools:

1. **Google Search**: a search engine that will search the internet to answer a question. This tool uses Serper API in the backend.
2. **Visit Tool**: visit a URL and render the page in HTML as in a browser.
3. **Page Up**: navigate the current page by scrolling up.
4. **Page Down**: navigate the current page by scrolling down.

Algorithm 3: HuggingFace Open Deep Research

Data: Task input x, LLM θ, maximum number of turns for search and main agents T_s and T_m, respectively, and planning interval p

web_tools $\leftarrow$ {Search, Visit, Page Up, Page Down, Finder, Find Next, Archive Search, Text Inspector};

main_tools $\leftarrow$ {search_agent, Visualize, Text Inspector};

Function *plan(q, c)*:
  /* Prompt the LLM to generate a plan */
  return $\theta(q, c; \textit{plan})$;

Function *search_agent(q)*:
  $P \leftarrow \text{plan}(q, \emptyset)$;
  $C \leftarrow \{q, P\}$;
  $t \leftarrow 1$;
  while $t < T_s$ **do**
    if $t \mod p = 0$ **then**
      $P \leftarrow \text{plan}(q, C)$;
      $C \leftarrow C \cup \{P\}$;
    $o_t \leftarrow \theta(C; \text{web\_tools})$;
    if $\textit{type}(o_t) = \textit{final\_answer}$ **then**
      return o_t;
    /* do the tool call, see A.2 for tool details */
    $C \leftarrow C \cup \{o_t, \text{tool}(o_t)\}$;
    $t \leftarrow t + 1$;
  return $\theta(C; \textit{final answer})$;

Result: Task output y
Turn $t \leftarrow 1$;
$P \leftarrow \text{plan}(x, \emptyset)$;
Context $C \leftarrow \{x, P\}$;
/* the main agent plans and calls the search agent */
while $t < T_m$ **do**
  if $t \mod p = 0$ **then**
    $P \leftarrow \text{plan}(x, C)$;
    $C \leftarrow C \cup \{P\}$;
  $o_t \leftarrow \theta(C; \text{main\_tools})$;
  if $\textit{type}(o_t) = \textit{final\_answer}$ **then**
    return o_t;
  $C \leftarrow C \cup \{o_t, \text{tool}(o_t)\}$;
  $t \leftarrow t + 1$;
return $\theta(C; \textit{final answer})$;

5. **Finder Tool**: find a text in the current page.

6. **Find Next**: find the next occurrence of the text in the current page.

Algorithm 4: GPT-Researcher

Data: Task input x, LLM θ, research depth D, research breadth B, summary interval n

Function *search(q)*:
  return $(\textit{title}_i, \textit{url}_i, \textit{snippet}_i)_{i=1}^{k}$;

Function *visit(u, q)*:
  $D \leftarrow \text{scrape}(u)$;
  $D \leftarrow \text{split}(D) = \{d_i\}_{i=1}^{m}$;
  if $q = \emptyset$ then **return** $d' \leftarrow d_1$;
  else $d' \leftarrow \arg\max_{d_i \in D} \text{F1}(d_i, q)$;
  return d';

Function *plan(q)*:
  /* Prompt the LLM to generate a list of queries */
  $R \leftarrow \text{search}(q)$;
  return $\theta(x, R; \textit{plan})$;

Function *conduct_research(q)*:
  /* Conduct research on one query by generating subqueries and retrieve and scrape */
  $Q \leftarrow \text{plan}(q)$;
  $R \leftarrow \emptyset$;
  for $q_i \in Q$ **do**
    for $t_i, u_i, s_i \in \textit{search}(q_i)$ **do**
      $r_i \leftarrow \text{visit}(u_i, s_i)$;
      $R \leftarrow R \cup r_i$;
  return $\theta(x, R; \textit{process})$;

Function *deep_research(q, d)*:
  /* Recursively plan and conduct research */
  $Q \leftarrow \text{plan}(q)$;
  $R \leftarrow \emptyset$;
  for $q_i \in Q$ **do**
    $r_i \leftarrow \text{conduct\_research}(q_i)$;
    /* Prompt the LLM to generate takeaways and follow up questions */
    $q_i' \leftarrow \theta(r_i; \text{process})$;
    if $d < D$ **then**
      $R \leftarrow R \cup \text{deep\_research}(q_i', d + 1)$;
  return R;

Result: Task output y
Turn $t \leftarrow 1$;
Context $C \leftarrow \{x\}$;
$P \leftarrow \text{plan}(x)$;
$R \leftarrow \text{deep\_research}(P, 1)$;
return $\theta(R; \textit{write report})$;

7. **Archive Search**: search the archives for information.

8. **Text Inspector**: given the path to a downloaded text file, it will call an LLM to answer questions about the text.

Detailed descriptions of each tool can be found in the original implementation[11].

A.3. Trajectory-Level Analysis Definitions

In this subsection, we describe how we annotate each trajectory with the failure modes. For LLM-as-a-judge approaches, we use `o3-2025-04-16` as the judge model. In each of the following LLM-as-a-judge approaches, we use the same judge model, and force the model to generate its response in a json format for easy parsing. We find that existing frontier LLMs are powerful enough to reliably check for simple yes/no questions and output them in a json format.

Confirmation bias. Confirmation bias occurs when the system finds a potential candidate that is incorrect in its search process, and subsequently spends the majority of its search budget on the same candidate without considering other options, leading to a lack of exploration in the search space. To detect this, we first collect all the search queries that the system has made and then use an LLM to check if the search queries overly focus on a single wrong candidate. The judge model is given access to the groundtruth answer and the search queries, so it's able to determine if the search queries are relevant to the groundtruth answer and the similarities between different search queries. We consider a trajectory to have confirmation bias if a majority of the search queries are similar to each other, and focuses on a single wrong candidate. The prompt used for confirmation bias detection is shown in Table 6.

Unfocused search. Unfocused search occurs when the system generates overly generic search queries that are not useful for narrowing down the search space—the system cannot make any progress towards finding useful information. To detect this, we first collect all the search queries that the system has made and then use an LLM to check if the search queries are generic and not useful for narrowing down the search space. We consider a trajectory to have unfocused search if a majority of the search queries are overly generic and not useful for narrowing down the search space. The prompt used for unfocused search detection is shown in Table 7.

Inefficient tool usage. Inefficient tool usage occurs when the system does not discover new information with its tool calls, and is therefore wasting its tool budget. Specifically, we use URLs as a proxy for the information discovered by the system—a tool call that only return URLs seen in previous search results is considered as a waste of tool budget. We use a simple heuristic for this analysis—iterate over all search calls made in the trajectory and keep track of seen URLs. Then, we report the percentage of search calls that only return URLs seen in previous search results.

[11] https://github.com/huggingface/smolagents/blob/main/src/smolagents/default_tools.py

Prompt for Confirmation Bias Detection
You are a helpful assistant that can analyze the trajectory of an informati question-answer pair and the search history of an agent that tried to answer t search history and determine if the agent spends more than half of the tool c answer. That is, the agent continues searching for the same topic even thou question, and spends half or more of its tool calls on these searches. Outp reasoning and a single word: 'yes' if the agent spends more than half of i answer or 'no' if the agent does not. **Reasoning:** explain what the agent did, and if it did or did not focus its sea **Conclusion:** "yes" or "no". **Search queries:** `<search-queries>` **Question:** `<question>` **Correct Answer:** `<correct-answer>`

Table 6. System prompt used for detecting confirmation bias in agent trajectories

Answer ignored. Answer ignored occurs when the system encounters the correct answer in its search process, but does not use it to answer the question. One possible explanation is that the system is distracted by other noisy information in its context, preventing it from correctly identifying the groundtruth. We employ a simple approach for this analysis—we check if the groundtruth answer is present in any of the tool responses. We employ a LLM judge to enable fuzzy matching between the groundtruth answer and the tool responses. The prompt used for answer ignored detection is shown in Table 8. We iterate over all tool calls and use this check to determine if any tool responses contain the groundtruth answer. We terminate the iteration if we find a tool response that contains the groundtruth answer, and report the percentage trajectories where at least one tool response contains the groundtruth answer.

Abstention. Abstention occurs when the system does not attempt to answer the question due to the lack of information in its context. Existing LLMs can often refuse to answer the question if it is not confident in answering the question, but this behavior is not desirable for search agents that could leverage additional tool calls to find the necessary information. We use a simple LLM judge to check if the system attempted to answer the question. The prompt used for giving up detection is shown in Table 9.

Hallucination. Hallucination occurs when the system gen-

Prompt for Unfocused Search Detection

You are a helpful assistant that can analyze the trajectory of an information seeking agent. You are given the question-answer pair and the search history of an agent that tried to answer the question. You should analyze the search history and determine if the search queries do not help the agent narrow down the search space in the following cases:
1. The agent searches for information relevant to the question and answer, but it's not specific enough to yield helpful results.
2. The agent searches for queries that are not sufficiently relevant to the question and answer, which does not narrow down the search space enough.
3. The agent explores the search space with diverse queries but does not use enough tool calls to properly narrow down the search space by either eliminating wrong answers or finding the correct answer.
All of these cases are considered to be unfocused search. You should consider the whole trajectory of the agent, and not just some of the tool calls—only consider the trajectory to be unfocused if most of the searches are unfocused.
Output your final conclusion with your reasoning and a single word: 'yes' if the searches are unfocused or 'no' if the searches are focused enough.
Reasoning: explain what the agent did, and if it did or did not use tool calls to properly narrow down the search space.
Conclusion: "yes" or "no".
Search queries: `<search-queries>`
Question: `<question>`
Correct Answer: `<correct-answer>`

Table 7. System prompt used for detecting unfocused search in agent trajectories

Prompt for Groundtruth Ignored Detection

You are a helpful assistant that can analyze the trajectory of an information seeking agent. You are given the question-answer pair and a list of webpages the agent read during its search process. You should analyze the web c that is either a direct or near-exact match to the correct answer. Output your f and a single word: 'yes' if the content contains the correct answer or 'no' correct answer.
Reasoning: explain if the web content contains the correct answer.
Conclusion: "yes" or "no".
Question: `<question>`
Correct Answer: `<correct-answer>`

Table 8. System prompt used for detecting groundtruth ignored in agent trajectories

Prompt for Giving Up Detection

You are a helpful assistant that can analyze the final output of an information the agent decides that it cannot find the correct answer. For example, if the e enough relevant information to answer the question, or if the response is sin the agent did not attempt to answer the question. Output your final conclus agent decides it did not find enough information to answer the question or "
Conclusion: "yes" or "no".
Final output: `<final-output>`

Table 9. System prompt used for detecting giving up in agent trajectories

erates information that is not supported by the information it has discovered in its search process. In agentic search systems, it is not desirable to hallucinate information, as it could result in incorrect and misleading answers and thus affect the trustworthiness of the system. Inspired by previous works(Rashkin et al., 2023; Bohnet et al., 2022; Gao et al., 2023), we check if the system hallucinates information by first decomposing the model's explanation into a set of atomic claims. Then, we iterate through all the tool responses from the search process and check if the tool responses support all the claims. As long as one tool response support a claim, we consider the system to not have hallucinated that claim. In the end, we report the average percentage of unsupported claims across trajectories. The prompt used for decomposing the model's explanation into a set of atomic claims is shown in Table 10, and the prompt used for hallucination detection is shown in Table 11. These prompts are derived from previous works that show LLMs can reliably decompose texts into a set of atomic claims and check if claims are supported by a piece of text—they also achieve high agreement with human judges (Gao et al., 2023; Kamoi et al., 2023; Yen et al., 2025).

A.4. SLIM Details and Ablations

We show an example of a SLIM trajectory in Figure 7. A sketch of the framework is also shown in Alg. 5. Furthermore, we ablate our design choices along the following

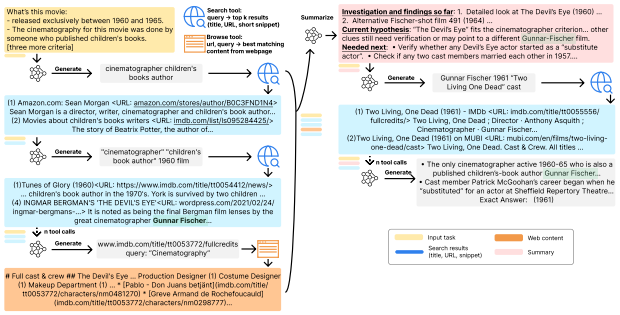

Figure 7. An example of a SLIM trajectory.

dimensions:

- **Summarization frequency:** Instead of summarizing the trajectory every $n = 50$ turns, we summarize every $n = 25$ turns.
- **Summarization trigger:** Instead of summarizing the trajectory every n turns, we summarize the trajectory when the input length exceeds a threshold $\tau = \{32768, 65536\}$ tokens.
- **Search tool:** We vary the number of top search results $k = \{10, 20\}$.
- **Browse tool:** We vary the maximum length of the scraped content $L = \{3000, 10000, 20000\}$ characters.

Prompt for Decomposing Explanation into Atomic Claims

Read the given explanation and generate a list of atomic claims that are supported by the explanation. Atomic claims that are basic facts that cannot be further broken down. Generate at most 10 claims for the explanation. Use the following as an example:

```
Explanation:  Searching UFCStats for featherweight bouts
where the loser landed 14 of 83 significant strikes (16.87 %)
and went 0-for-4 on takedowns returns the fight Myles Jury
vs. Ricky Glenn at UFC 219: Cyborg vs Holm (30 Dec 2017).
• Ricky Glenn (nickname "The Gladiator"|a synonym
for swordsman)
was the loser: sig. strikes 14/83 (16.87 %), takedowns 0/4.
• Both fighters (Jury 29, Glenn 28) were under 35 and
are American.
• The referee was John McCarthy, whose first event for
the UFC was in 1994.
Thus, the MMA event is UFC 219: Cyborg vs Holm.

Exact Answer: UFC 219: Cyborg vs Holm

Confidence: 75%
```

Atomic Claims:
- Ricky Glenn was the loser
- Ricky Glenn was nicknamed "The Gladiator"
- The sig. strike rate of Ricky Glenn was 14/83 (16.87- The takedown rate of Ricky Glenn was 0/4
- Jury was age 29
- Glenn was age 28
- Jury is American
- Glenn is American
- The referee was John McCarthy
- John McCarthy's first event for the UFC was in 1994

Output the atomic claims in the form of a json list.

Table 10. System prompt used for decomposing the model's explanation into a set of atomic claims

Prompt for Hallucination Detection

You are supported by the explanation. Atomic claims are used to identify the trajectory of an information of webpages and 10 claims for the explanation. You should analyze the claim is supported by the web content. A claim is supported by the web c mostly supported by the web content, and is not contradicted by the web co with a list of claims that are supported by the web content. Output the list only need to write the index of the supported claims in the list and nothing

Webpages: <webpages>

Atomic Claims: <atomic-claims>

Table 11. System prompt used for detecting hallucination in agent trajectories

Algorithm 5: SLIM

Data: Task input x, LLM θ, maximum number of turns T, summary interval n

Function *search(q)*:
  return $(title_i, url_i, snippet_i)_{i=1}^{k}$;

Function *browse(u, q)*:
  $D \leftarrow \text{scrape}(u)$;
  $D \leftarrow \text{split}(D) = \{d_i\}_{i=1}^{m}$;
  if $q = \emptyset$ then **return** $d' \leftarrow d_1$;
  else $d' \leftarrow \arg\max_{d_i \in D} \text{ROUGE-L}(d_i, q)$;
  return d';

Result: Task output y
Turn $t \leftarrow 1$;
Context $C \leftarrow \{x\}$;
$\mathcal{T} \leftarrow \{\text{search}, \text{browse}\}$;
while $t < T$ **do**
  if $t \mod n = 0$ **then**
    $C \leftarrow \theta(C; \text{summarize})$; /* Summarize every n turns */
  $o_t \leftarrow \theta(C; \mathcal{T})$;
  switch o_t **do**
    case *search* **do**
      $q_t \leftarrow o_t$;
      $C \leftarrow C \cup \{o_t, \text{search}(q_t)\}$;
    case *browse* **do**
      $u_t, s_t \leftarrow o_t$;
      $C \leftarrow C \cup \{o_t, \text{browse}(u_t, s_t)\}$;
    case *Final Answer* **do**
      return o_t;
  $t \leftarrow t + 1$;
return $\theta(C; \textit{final answer})$;

We also ablate the chunking and scoring strategy. By default, we chunk by natural paragraphs (splitting at newlines) and use ROUGE-L as the similarity metric. We also try using BM25 (Robertson & Zaragoza, 2009) as the similarity metric and splitting the content into chunks of 100 words (splitting at any whitespace).

For these ablations, we use o4-mini as the base model due to its cheaper cost and test on a smaller subset of 50 samples for each dataset. The results are shown in Table 12.

A.5. Experimental Details

We use o3, o4-mini, GPT-OSS-120B, GLM-4.7 Flash, Tongyi-DeepResearch-30B, and Claude-4-Sonnet as our base models. To calculate the cost, we use the prices listed in Table 13, which are obtained from respective websites https://platform.openai.com/docs/models/o3, https://platform.openai.com/docs/models/o4-mini, https://www.together.ai/models/gpt-oss-120b, https://openrouter.ai/z-ai/glm-4.7-flash, https://openrouter.ai/

Table 12. Ablation results with o4-mini as the base model. The number of tokens is shown in 10,000s. The cost is shown in US dollars. We ablate design choices in the summarization module, chunking strategy, and search and browse tool. For all settings, we set the tool budget to 100. The default setting summarizes every $n = 50$ turns, chunks by newline, use ROUGE-L as the similarity metric, and search returns the top $k = 10$ search results while browsing returns at most $L = 10,000$ characters. These experiments use a smaller subset of 100 samples for each dataset, so they are not directly comparable to the main results. Each experiment is run with three random seeds and the results are the mean and standard deviation.

	BrowseComp				HLE			
	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)
SLIM Default	40.67±5.86	118.27±5.93	54.02±2.43	1.33±0.07	17.33±3.06	11.39±1.34	7.61±0.46	0.13±0.01
Summarization Module								
$n = 25$	30.33±4.51	57.64±4.87	35.47±1.04	0.65±0.05	21.67±4.04	8.53±1.68	6.09±0.96	0.1±0.02
Summarize at 32K tokens	29.67±2.89	46±3.23	32.7±0.79	0.53±0.04	17.67±6.51	9.22±1.34	6.62±0.69	0.11±0.02
Summarize at 64K tokens	42.67±2.08	126.2±5.69	57.23±2.14	1.42±0.06	19.67±4.51	11.67±0.74	7.83±0.34	0.13±0.01
Chunking								
Split newline, BM25	37.67±3.51	121.38±8.56	55.3±2.01	1.37±0.1	21.33±4.04	12.21±1.14	8.33±0.31	0.14±0.01
Split words, ROUGE	39.33±5.03	113.55±2.88	52.97±1.39	1.28±0.03	19.33±5.69	11.69±1.05	7.91±0.44	0.13±0.01
Split words, BM25	40.67±2.52	121.39±3.33	55.95±1.18	1.37±0.04	20.33±4.16	10.4±0.66	7.32±0.53	0.12±0.01
Search and Browse								
No visit	34.33±2.89	111.53±5.1	63.47±2.03	1.26±0.06	15.33±1.15	14.35±1.68	10.42±0.9	0.16±0.02
No query in visit	37.33±1.15	187.63±9.54	66.82±2.34	2.1±0.11	20.33±2.08	14.55±0.75	8.9±0.62	0.17±0.01
$k = 10, L = 3,000$	42±6.24	111.59±12.51	52.77±3.95	1.26±0.14	21.33±1.53	11.65±1.2	7.75±0.07	0.13±0.01
$k = 10, L = 20,000$	38.67±1.53	117.35±7.79	54.5±1.53	1.32±0.09	20.67±0.58	12.19±1.24	7.84±0.57	0.14±0.01

alibaba/tongyi-deepresearch-30b-a3b, https://claude.com/pricing#api, https://www.firecrawl.dev/pricing.

For all models, we use a temperature of 1.0 and a maximum output token of 32,768. For o3 and o4-mini, we always use the default reasoning effort of "medium" and for Claude-4-Sonnet, we set the maximum number of thinking tokens to 30,000.

To calculate the token cost, we take a weighted sum of the token usage across all LLM calls: non-cached input tokens plus the total output tokens, and multiply the results by price per token (which is different for input and output tokens). We exclude cached tokens from the calculation because in practice, long-horizon systems are expected to have a large amount of cached tokens and system implementation that takes advantage of caching. Then, for the total cost, we add in the number of search API and scrape URL operations, multiplied by their respective prices. For the number of tool calls, we count the number of times the search API and scrape operations, the two atomic tool operations, are called.

We also include the results of other trained systems in Table 14. For OpenAI Deep Research (DR), the HLE number from the original blog post[12] and the BrowseComp number is from the BrowseComp paper (Wei et al., 2025). For Grok-4, the HLE number is from the original Grok 4 blog post [13] and the BrowseComp number is from the Grok 4 Fast blog post [14]. The WebResearcher (WebR) numbers are from the original paper (Qiao et al., 2025), where we show the results of the main WebResearcher-30B-A3B model; we exclude the heavy version since it uses multiple samples and aggregate the results. The WebThinker (WebT) numbers are from the original paper (Li et al., 2025c), where we show the results of the main WebThinker-32B model. They did not evaluate on BrowseComp, so we only report the HLE number.

[12] https://openai.com/index/introducing-deep-research/

[13] https://x.ai/news/grok-4

[14] https://x.ai/news/grok-4-fast

Table 13. Pricing for different components. Numbers are obtained from respective websites: open-weight models use pricing from OpenRouter.

	Cost
o3	\$2.0 / M input token, \$8.0 / M output token
o4-mini	\$1.1 / M input token, \$4.4 / M output token
Claude-4-Sonnet	\$3.0 / M input token, \$15.0 / M output token
GPT-OSS-120B	\$0.15 / M input token, \$0.6 / M output token
Tongyi-DeepResearch	\$0.09 / M input token, \$0.45 / M output token
GLM-4.7 Flash	\$0.07 / M input token, \$0.4 / M output token
Google search	\$0.5 / K query
Scrape URL	\$0.83 / K query

A.6. Additional Results

Main Results. We show the results of SLIM with o3 as the base model over three random seeds in Table 16. Here we also provide the concrete results for SLIM with different base models—Claude-4-Sonnet is shown in Table 17. For BrowseComp, HLE, and GAIA, we report the final correctness score. For DeepSearchQA, we report the fully correct score. For HealthBench Hard, we report the overall score. Due to API costs, we only evaluate Claude-4-Sonnet and Tongyi-DeepResearch-30B on BrowseComp and HLE.

We show additional results for GPT-OSS-120B and GLM-4.7 Flash in Table 15. For GLM-4.7 Flash, we also test using discard as a context management strategy, where the entire context is discarded when the rollout reaches the summarization step (i.e., 50 turns in our settings). We find that summarization performs substantially better on BrowseComp and DeepSearchQA and only marginally worse on HLE, GAIA, and HealthBench Hard, thus, we chose to use summarization for the main experiments. For GPT-OSS-120B, we also report the results for SLIM with a tool budget of $T = 100$ so the total costs are more comparable with the baselines.

REACT Ablations. We vary the number of search results k and the maximum length of the scraped content L for REACT to see the effect of search tool design choices, as shown in Table 18. We found that overall there aren't significant differences in the HLE results, but using fewer search results $k = 5$ than the default $k = 10$ leads to a 2.7 points

Table 14. Main results with o3 and o4-mini as the base model. All results are macro-averaged across test instances. The number of tokens is shown in 10,000s. The cost is shown in US cents. T denotes the tool budget. For reference only, † marks deep research systems that underwent task-specific training. Numbers are from the original reports (OpenAI, 2025; xAI, 2025; Qiao et al., 2025; Li et al., 2025c), and are not directly comparable due to different subsets of test instances used.

	Model	T	BrowseComp			HLE			DeepSearchQA			GAIA			HealthBench Hard		
			Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score
o3	Base	-	0.0	7.6	17.0	0.0	5.3	18.3	0.0	2.2	20.7	0.0	2.4	35.0	0.0	0.9	35.9
	SLIM	10	9.6	6.6	22.0	4.1	9.2	25.7	9.6	6.8	21.7	7.4	6.6	53.4	1.4	1.8	42.4
		25	21.5	20.8	33.0	7.3	18.6	29.7	22.0	24.2	35.0	13.4	17.2	67.0	1.6	2.0	41.3
		50	36.7	55.5	44.7	9.4	27.6	30.7	37.1	74.9	56.0	20.2	39.3	69.9	1.6	2.1	38.5
		100	47.3	105.7	53.7	9.3	30.6	31.7	49.5	90.9	59.0	24.7	49.8	74.8	1.8	2.1	42.0
		150	75.9	124.6	**56.0**	11.4	36.2	**33.3**	60.5	114.1	**62.0**	23.6	47.5	**77.7**	1.6	2.0	**42.5**
	REACT	1	9.0	7.9	4.3	5.7	9.7	16.0	1.0	1.3	17.0	0.9	1.7	30.1	2.9	4.5	39.8
		5	19.2	14.8	6.7	10.0	12.3	19.7	4.1	2.7	20.7	3.5	2.9	30.1	2.8	3.7	39.2
		10	22.9	17.9	7.0	10.9	14.9	21.3	6.9	4.6	17.3	6.0	4.4	35.9	3.0	4.2	37.1
	SEARCH-O1	1	9.5	8.8	18.0	5.2	7.5	20.0	10.1	5.4	28.7	8.5	5.0	50.5	1.2	1.4	39.2
		5	46.9	22.2	24.0	18.7	13.7	20.7	46.1	17.7	32.3	28.4	11.8	65.0	2.2	1.8	37.8
		10	89.8	39.2	31.0	23.9	16.7	26.3	84.1	32.6	35.0	45.8	18.6	63.1	2.5	1.8	37.6
		25	183.2	79.8	40.0	44.2	28.5	25.0	169.9	70.1	40.0	79.0	34.8	67.0	2.2	1.8	39.5
		50	306.2	143.3	48.3	49.8	32.6	27.0	248.2	113.1	45.3	109.9	52.8	67.0	2.6	1.9	37.4
		100	456.7	246.7	55.7	52.2	37.1	27.0	353.5	189.7	44.0	141.7	79.1	69.9	2.5	1.8	38.5
	HF-ODR	-	8.4	59.5	20.0	1.7	18.5	17.7	6.8	52.8	30.7	5.1	45.2	51.5	1.4	6.2	32.1
	GPT-OSS-120B	-	69.5	29.3	10.7	85.6	31.9	16.0	47.6	20.2	23.3	48.8	21.0	45.6	104.6	38.7	16.3
o4-Mini	Base	-	0.0	5.7	5.0	0.0	2.5	15.0	0.0	1.6	14.0	0.0	1.8	28.2	0.0	0.5	18.2
	SLIM	10	9.8	7.7	12.7	3.4	4.8	20.7	9.7	5.4	19.7	7.2	4.5	50.5	1.5	1.1	23.0
		25	22.8	27.4	21.3	5.4	8.0	24.0	22.6	21.1	24.3	14.6	14.5	59.2	2.5	1.9	22.9
		50	40.3	84.8	31.0	7.1	12.2	25.7	40.1	65.2	37.7	20.5	29.0	65.0	2.6	2.2	22.7
		100	48.5	116.8	33.3	7.2	12.9	**26.7**	60.6	94.4	41.3	23.5	32.3	**71.8**	2.4	2.2	22.3
		150	84.1	169.5	**37.7**	7.8	13.7	25.3	73.8	116.8	**46.3**	25.7	36.8	67.0	2.4	1.8	22.4
	REACT	1	9.3	5.8	1.3	4.6	4.8	17.0	1.0	0.6	7.0	0.8	0.9	21.4	1.6	1.6	22.1
		5	18.9	10.0	3.0	6.4	5.5	15.3	4.7	2.2	10.7	3.8	2.3	29.1	2.1	1.9	20.8
		10	20.6	9.8	2.3	6.9	5.9	15.3	7.5	3.8	12.0	5.7	3.3	32.0	2.4	1.9	21.6
	SEARCH-O1	1	10.0	8.1	6.3	3.5	3.4	13.0	9.7	3.8	15.7	8.0	3.4	42.7	0.7	0.7	20.1
		5	49.7	21.3	11.3	11.9	6.1	23.3	45.2	13.9	27.3	29.9	9.1	57.3	1.8	1.0	20.8
		10	93.9	36.3	17.3	15.6	7.3	17.0	79.8	25.4	31.3	41.9	13.7	54.4	1.7	1.0	**23.2**
		25	207.7	74.8	25.0	22.5	9.2	22.3	130.6	40.4	32.3	22.2	12.2	34.0	2.7	1.2	20.5
		50	351.5	126.0	28.7	26.3	11.6	19.3	226.3	70.2	35.7	82.1	24.3	59.2	1.7	0.9	21.6
		100	546.7	201.6	36.0	25.8	10.8	21.3	279.4	89.2	34.0	83.6	25.9	55.3	2.0	1.1	19.6
	HF-ODR	-	15.4	52.1	15.0	3.9	12.8	16.3	17.1	77.0	34.7	11.7	49.2	62.1	6.0	19.9	-0.1
	GPT-R	-	82.5	29.5	4.0	100.8	32.8	11.3	56.3	22.6	15.7	62.9	23.1	42.7	112.3	35.6	-0.2
Others	OpenAI DR†	-	-	-	51.5	-	-	26.6	-	-	-						
	Grok-4†	-	-	-	43.0	-	-	38.6	-	-	-						
	WebR-30B†	-	-	-	37.3	-	-	28.8	-	-	-						
	WebT-32B†	-	-	-	15.8	-	-	-	-	-	-						

	Model	T	BrowseComp			HLE			DeepSearchQA			GAIA			HealthBench Hard		
			Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score	Tools	Cost	Score
GLM-4.7	Base	-	0.0	0.7	1.3	0.0	0.7	9.7	0.0	0.4	7.3	0.0	0.5	14.6	0.0	0.1	6.8
	REACT	10	27.8	3.0	2.0	17.8	2.2	9.0	30.1	2.9	8.0	24.7	2.6	18.4	15.6	1.5	12.9
	SEARCH-O1	50	155.3	15.9	7.7	49.4	5.3	9.7	116.0	11.7	19.0	69.0	7.0	45.6	23.0	2.3	11.5
	SLIM	150	51.8	4.9	**17.3**	19.8	2.5	13.7	48.8	5.1	**20.7**	26.0	2.7	49.5	5.2	0.5	13.7
	SLIM-Discard	150	51.5	4.6	14.0	19.1	2.3	**14.7**	53.1	5.6	14.3	27.5	2.9	**50.5**	5.0	0.4	**14.5**
GPT-OSS	Base	-	0.0	0.2	2.7	0.0	0.2	10.0	0.0	0.1	9.3	0.0	0.1	23.3	0.0	0.1	36.6
	REACT	10	26.1	3.2	3.3	10.3	1.4	11.0	34.8	4.3	4.7	25.0	3.1	30.1	4.8	0.8	39.1
	SEARCH-O1	50	45.7	4.4	6.0	9.7	1.1	16.0	60.1	5.8	15.0	29.1	2.9	45.6	2.6	0.4	37.6
		100	60.5	10.6	**20.0**	7.8	1.3	17.0	51.3	9.8	34.0	19.6	3.0	55.3	2.2	0.4	38.6
	SLIM	150	70.5	12.5	20.0	8.1	1.3	**19.0**	61.6	11.6	**34.3**	23.5	3.9	**58.3**	2.2	0.4	**39.3**

Table 15. Additional results for GPT-OSS-120B and GLM-4.7 Flash. Costs are in US cents. For GLM-4.7 Flash, we also test using discard as a context management strategy (SLIM-Discard). For GPT-OSS-120B, we also report the results for SLIM with a tool budget of $T = 100$ so the total costs are more comparable with the baselines.

Table 16. Statistical significance analysis with o3 as the base model. We run with three random seeds for each experiment and report the mean and standard deviation.

		BrowseComp				HLE			
		Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)
o3	-	17.22±1.02	3.87±0.12	0±0	0.08±0	19.56±1.07	2.63±0.04	0±0	0.05±0
Search-o1	50	49.33±1.2	**49.98±1.5**	298.9±6.84	1.24±0.04	26.78±0.69	**13.05±0.52**	50.96±1.17	**0.3±0.01**
SLIM	150	**53±1.2**	54.77±5.23	**50.84±0.44**	**1.12±0.1**	**32.11±1.84**	16.44±1.15	**10.3±0.98**	0.33±0.02

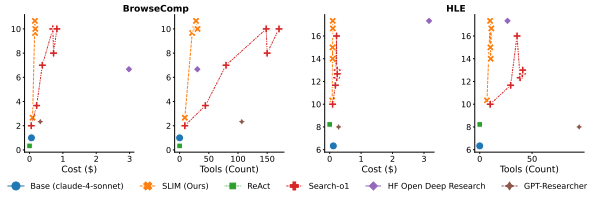

Figure 8. With Claude-4-Sonnet as the base model, SLIM consistently outperforms other baselines on BrowseComp while using fewer tool calls and lower overall costs. On HLE, SLIM can achieve overall higher performance and use fewer tool calls.

Table 17. Main results with Claude-4-Sonnet as the base model. All results are macro-averaged across test instances. The number of tokens is shown in 10,000s. The cost is shown in US dollars. T denotes the maximum number of turns in each trajectory.

	T	BrowseComp				HLE			
		Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)
Claude-4-Sonnet	-	1.0	1.9	0.0	0.06	6.3	3.9	0.0	0.12
REACT	1	0.3	0.0	0.0	0.00	8.3	0.0	0.0	0.00
	5	0.3	0.0	0.0	0.00	8.3	0.0	0.0	0.00
	10	0.3	0.0	0.0	0.00	8.0	0.0	0.0	0.00
SEARCH-O1	1	2.0	1.5	9.0	0.05	10.0	2.9	10.0	0.09
	5	3.7	6.0	44.1	0.21	11.7	5.3	29.5	0.18
	10	7.0	10.7	79.5	0.38	16.0	6.3	35.6	0.22
	25	8.0	20.1	149.9	0.72	13.0	6.8	41.1	0.24
	50	10.0	22.9	170.3	0.82	12.7	7.0	40.7	0.24
	100	10.0	19.4	148.3	0.70	12.3	6.4	38.5	0.22
HF-ODR	20	6.7	98.8	30.4	2.98	17.3	105.0	26.5	3.16
GPT-R	-	2.3	7.9	106.5	0.32	8.0	6.9	94.9	0.28
SLIM	10	2.7	2.8	8.9	0.09	10.3	2.5	6.9	0.08
	25	9.7	5.1	21.6	0.17	15.0	2.8	10.2	0.09
	50	10.0	5.0	27.1	0.16	17.3	3.0	9.9	0.10
	100	**10.7**	4.8	28.1	0.16	14.0	2.9	10.5	0.09
	150	10.0	5.2	30.7	0.17	**16.7**	3.1	11.1	0.10

improvement in the BrowseComp results. This is likely due to the fact that search results lower in the ranking are often noisy and irrelevant to the question, and using fewer but more relevant search results leads to a more focused search process. Furthermore, fewer search results means less context is added to the LLM, preventing it from hitting the context window limit as much. This is evident in more token and tool usage. However, we use $k = 10$ for the main experiments to stay consistent with the other baselines.

Table 18. REACT ablations with o3 as the base model, and the maximum number of turns is $T = 10$. We vary the number of search results k and the maximum length of the scraped content L.

	Parameters			BrowseComp				HLE			
	T	k	L	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)	Score (↑)	Tokens (↓)	Tools (↓)	Cost (↓)
REACT	10	10	10k	7.0	8.0	2.8	0.16	21.3	7.0	1.2	0.14
REACT	10	5	10k	9.7	10.6	4.1	0.21	21.7	7.0	1.7	0.14
REACT	10	10	3k	5.0	8.7	2.8	0.18	22.7	6.5	1.2	0.13
REACT	10	5	3k	8.3	10.7	4.1	0.22	21.3	6.7	1.7	0.13

A.7. Additional Analysis

In this subsection, we provide additional analysis—we extend the initial outcome-based analysis to SLIM, and show the trajectory-level analysis on the more comprehensive baselines.

In Table 19, we show the trajectory-level analysis where we report the failure modes as a percentage of trajectories that ends with an incorrect answer. The trends are consistent with the analysis in the main text, but we find that SLIM can often find the correct answer across its long trajectories—over 69% of the incorrect trajectories encounters the correct answer, but the model is not able to identify and use it to answer the question. This could be attributed to the fact that modern LLMs still struggle at long-context settings where it may need to reason over many sources. We leave these improvements to future work.

Table 19. For correct, we report the percentage of trajectories across all samples. For each trajectory-level failure mode, we report the percentage of trajectories that ends with an incorrect answer. For hallucination only, we report the percentage of hallucinations for samples that ends with an incorrect answer and do not abstain.

Framework	Turn Budget	Correct	Confirm Bias	Unfocused Search	Inefficient Search	Abstention	Answer Ignored	Hallucinate
REACT	10	7.0	10.0	47.3	4.2	1.1	0.7	56.7
SEARCH-O1	50	48.3	18.1	65.2	14.0	8.4	50.3	46.8
HF-ODR	20	20.0	8.6	75.5	56.5	41.6	2.1	96.2
SLIM	150	56.0	22.0	77.3	17.2	62.9	69.7	19.0