VERISEARCH: A VERIFICATION-ENHANCED SEARCH AGENT WITH PROACTIVE EVIDENCE FILTERING AND CONFLICT RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Search agents built on large language models (LLMs) answer knowledge-intensive questions by retrieving and reading webpages during inference. This allows them to supplement static parametric memory with external evidence. Existing agents primarily optimize search execution, while retrieved webpages may not provide sufficient information for the current question, and different sources may present conflicting claims. These issues can contribute to error propagation across subsequent reasoning steps. We propose VeriSearch, a verification-enhanced search agent that introduces explicit evidence control into the search workflow. Before full-page crawling, we design AnswerabilityPruner, which estimates the answerability of each search result from its search snippet and triggers a reflective retry when no candidate passes the answerability threshold. After crawling, ConflictResolver detects cross-source factual conflicts among retrieved evidence and resolves them through weighted evidence arbitration. With Qwen3.5-9B as the backbone, VeriSearch outperforms Search-o1 by an average of 10.3 points across three benchmarks and exceeds Claude-4-Sonnet's reported results by an average of 5.0 points on xbench-DeepSearch and SEAL-0. Code is included in the supplementary material and will be made publicly available upon publication.

1 INTRODUCTION

Large language models (LLMs) exhibit strong reasoning capabilities, yet their performance on knowledge-intensive tasks remains constrained by static parametric memory. Consequently, they may struggle with long-tail knowledge and temporally evolving information (Kandpal et al., 2023; Dhingra et al., 2022; Mallen et al., 2023). Search agents address this limitation by combining LLM reasoning with web search to retrieve external information during inference (Huang et al., 2025; Li et al., 2026).

However, retrieval from the open web introduces additional challenges. Search results may correspond to webpages that lack sufficient information for the current sub-question (Li et al., 2025). Different sources may also provide conflicting claims about the same fact (Ge et al., 2025). Following the taxonomy of Xu et al. (2024), conflicts among retrieved webpages can be viewed as a retrieval-specific form of inter-context conflict. We refer to such conflicts as cross-source factual conflicts. Once erroneous intermediate information enters iterative retrieval and reasoning, it may propagate to subsequent reasoning steps (Wang et al., 2026).

Existing search agents have primarily improved how information is searched and retrieved. Early paradigms established foundational reasoning and tool-use workflows for search agents (Yao et al., 2022; Nakano et al., 2021; Schick et al., 2023). Building on these foundations, existing work has improved the search process through several directions, including task planning (Mei et al., 2025; Qian et al., 2026), tool invocation (Jin et al., 2025a;b), iterative search (Gao et al., 2025), and deep web exploration (Li et al., 2026).

However, these lines of work primarily focus on improving the search process itself and do not jointly address two practical issues that arise when searching the open web: estimating the answerability of each search result from its search snippet before full-page crawling and resolving cross-source factual conflicts among retrieved evidence. Without explicit handling of these issues,

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

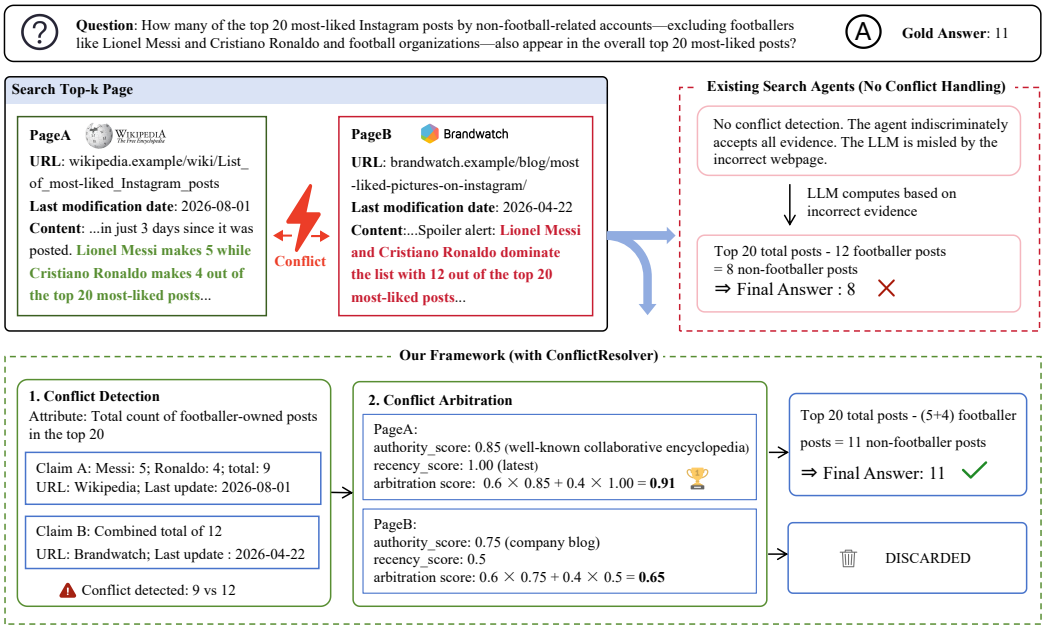

Figure 1: A representative case showing how unresolved cross-source factual conflicts can affect downstream reasoning. Wikipedia (Page A) and Brandwatch (Page B) report 9 and 12 football-related posts, respectively. Without conflict resolution, both claims enter the reasoning context and lead to an incorrect computation. ConflictResolver detects the conflict and performs weighted evidence arbitration. Page A receives a higher arbitration score than Page B, so VeriSearch retains the evidence from Page A and discards that from Page B, yielding the correct answer of 11.

evidence from low-answerability search results or unresolved conflicting evidence may enter the reasoning context and lead to error propagation across subsequent reasoning steps.

As illustrated in Figure 1, unresolved cross-source factual conflicts may affect downstream reasoning. When conflicting claims are incorporated into the same reasoning context without explicit resolution, they can provide incompatible premises and increase the risk of reasoning errors. In the example, this results in an incorrect computation. VeriSearch instead detects the cross-source factual conflict and applies weighted evidence arbitration before answer generation, yielding the correct answer in this example.

Addressing these issues presents two challenges. First, estimating the answerability of a candidate search result from its search snippet remains challenging. Search snippets are often sparse, truncated, and query-dependent, which limits the information available for answerability estimation. Second, resolving cross-source factual conflicts remains challenging without direct access to ground truth. Conflicting evidence may differ in the strength of its support and source characteristics, making direct adjudication difficult.

To address these challenges, we propose VeriSearch, a verification-enhanced search agent that augments the standard workflow of planning, retrieval, and reading (Huang et al., 2025; Qin et al., 2026). VeriSearch retains Planner and Searcher for question decomposition and web retrieval, while introducing AnswerabilityPruner and ConflictResolver for answerability-based filtering and cross-source factual conflict resolution.

We summarize our main contributions as follows:

- We propose VeriSearch, a search agent framework that augments the standard iterative search workflow with dedicated mechanisms for answerability-based filtering and cross-source factual conflict resolution.

- We introduce AnswerabilityPruner, which assigns an answerability score to each search result based on its search snippet before full-page crawling, filters candidates using an answerability threshold, and triggers reflective retries when no candidate meets the threshold.
- We introduce ConflictResolver, which detects cross-source factual conflicts and resolves them through weighted evidence arbitration using estimated source authority and webpage recency as heuristic arbitration signals.

Compared with Search-o1 (Li et al., 2025) using the same Qwen3.5-9B backbone, VeriSearch improves by 15.3, 14.5, and 1.1 points on xbench-DeepSearch, SEAL-0, and Bamboogle, respectively. VeriSearch also exceeds Claude-4-Sonnet's reported results by 2.7 points on xbench-DeepSearch and 7.3 points on SEAL-0.

2 VERISEARCH

Overview. Given a user query Q, VeriSearch operates iteratively through four modules, as illustrated in Figure 2:

- **Planner (§2.1):** The Planner iteratively decomposes Q into atomic sub-questions q_t. After each sub-answer a_t, it decides whether to continue planning or aggregate the accumulated sub-answers into the final answer A.
- **Searcher (§2.2):** For each q_t, the Searcher rewrites it into a set of search queries $\widetilde{\mathcal{Q}}_t$ and executes them through the search engine, yielding deduplicated search results $\mathcal{W}_t$. During reflective retries, query rewriting is additionally conditioned on the accumulated failure history.
- **AnswerabilityPruner (§2.3):** AnswerabilityPruner estimates the answerability of each $w_i \in \mathcal{W}_t$ from its search snippet. Search results that pass the answerability threshold are crawled, and the cleaned webpages are consolidated into the evidence set E_t.
- **ConflictResolver (§2.4):** ConflictResolver detects cross-source factual conflicts within E_t and resolves them through weighted evidence arbitration using estimated source authority and webpage recency as heuristic arbitration signals. It produces the processed evidence set E'_t and the sub-answer a_t.

2.1 PLANNER: QUESTION DECOMPOSITION AND ANSWER AGGREGATION

Unlike methods that generate a complete static plan before retrieval (He et al., 2025), the Planner adopts a dynamic decomposition strategy. At each planning iteration t, it generates a single atomic sub-question q_t conditioned on the original query Q and the sub-answers $\{a_1, \ldots, a_{t-1}\}$ accumulated from previous iterations. The LLM also assesses whether these sub-answers provide sufficient information to answer Q. Once they are deemed sufficient, or the maximum number of planning iterations $T_{\max}$ is reached, the Planner aggregates them into the final answer A.

2.2 SEARCHER: UNIFIED SEARCH INTERFACE

The Searcher serves as the sole interface between VeriSearch and the open web.

Query Rewriting. The atomic sub-question q_t is typically phrased as a natural-language question and may contain pronouns without explicit entity references, making it unsuitable for direct keyword search. The Searcher therefore rewrites q_t into a set of keyword-style queries $\widetilde{\mathcal{Q}}_t = \{\tilde{q}_\ell\}_{\ell=1}^{L}$, conditioned on Q for entity disambiguation. Here, L denotes the number of generated queries. During reflective retries, query rewriting is additionally conditioned on the accumulated failure history, as described in Section 2.3.

Parallel Search and Deduplication. The Searcher invokes the search engine API in parallel over $\widetilde{\mathcal{Q}}_t$. The returned lists are deduplicated and merged by URL, yielding the search results $\mathcal{W}_t = \{w_i\}_{i=1}^{M}$, where M denotes the number of deduplicated results. Each result is represented as a quadruple $w_i = (h_i, u_i, d_i, r_i)$, where h_i denotes the page title, u_i the URL, d_i the snippet text, and r_i the search-engine rank.

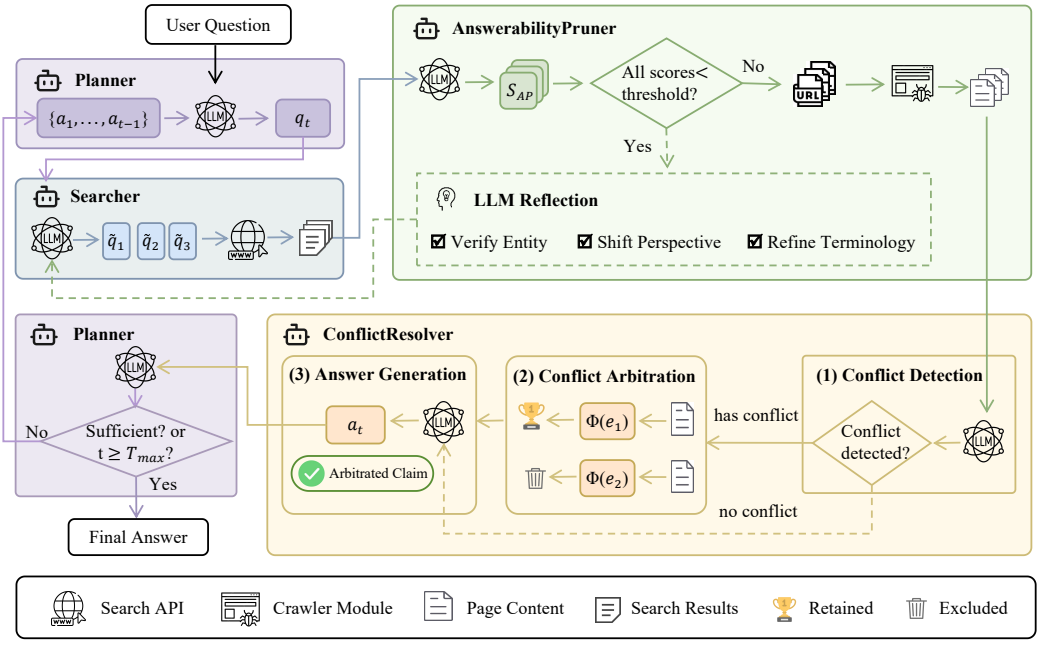

Figure 2: **Overview of the VeriSearch framework.** VeriSearch iterates through Planner, Searcher, AnswerabilityPruner, and ConflictResolver. AnswerabilityPruner filters search results before full-page crawling, while ConflictResolver resolves cross-source factual conflicts through weighted evidence arbitration and then generates the sub-answer a_t.

2.3 AnswerabilityPruner: Answerability-Based Filtering and Reflective Retry

AnswerabilityPruner performs answerability scoring over snippets after search results are obtained but before full-page crawling. It then applies answerability-based filtering to determine which results are crawled. We define answerability as the estimated degree to which the corresponding full webpage is likely to contain sufficient information to answer the current sub-question.

Answerability Scoring and Threshold Gating. Crawling all retrieved results may introduce irrelevant content into the reasoning context. AnswerabilityPruner therefore assigns an answerability score to each $w_i \in \mathcal{W}_t$, conditioned on both q_t and Q:

$$S_{\mathrm{AP}}(w_i) = \mathrm{LLM}_{\mathrm{AP}}(q_t, Q, w_i), \tag{1}$$

where $\mathrm{LLM}_{\mathrm{AP}}$ denotes the LLM-based evaluator used by AnswerabilityPruner, and $S_{\mathrm{AP}}(w_i) \in [0, 10] \cap \mathbb{Z}$ denotes the answerability score assigned to search result w_i. Only results satisfying $S_{\mathrm{AP}}(w_i) \geq \theta$ are selected for full-page crawling.

Reflective Retry Mechanism. As illustrated by the light green AnswerabilityPruner block in Figure 2, if $\max_{w_i \in \mathcal{W}_t} S_{\mathrm{AP}}(w_i) < \theta$, no candidate passes the answerability threshold and the reflective retry mechanism is triggered. The mechanism allows at most K retry attempts. Let $\mathcal{H}_{\mathrm{fail}}^{(k-1)}$ denote the failure history accumulated before the k-th retry. It contains the queries from previous search attempts, together with the titles and URLs of their returned results. The LLM then reflects on these failures along three dimensions: entity verification, perspective shifting, and terminology refinement. This reflection guides the Searcher to generate a revised query set:

$$\widetilde{\mathcal{Q}}_t^{(k)} = \mathrm{LLM}_{\mathrm{rewrite}}\left(q_t, Q, \mathcal{H}_{\mathrm{fail}}^{(k-1)}\right). \tag{2}$$

The Searcher then submits $\widetilde{\mathcal{Q}}_t^{(k)}$ to the search engine, and AnswerabilityPruner scores the returned results using S_{AP}. This reflection-retrieval loop continues for at most K retries. If no result satisfies

$S_{\mathrm{AP}}(w_i) \geq \theta$, the result with the highest answerability score across all search attempts is selected for crawling as the fallback.

Evidence Construction. For each search result that passes the answerability threshold, the framework fetches the corresponding full webpage and cleans its content. It also extracts the webpage's last update date for use as a recency signal in ConflictResolver. Each evidence item is represented as a triple $e_j = (c_j, u_j, \tau_j)$, where c_j denotes the webpage content, u_j the source URL, and τ_j the last update date. The originating search-engine rank is retained as auxiliary metadata for the tie-breaking rule described below. The resulting evidence set is $E_t = \{e_j\}_{j=1}^{N}$, where N denotes the number of evidence items.

2.4 CONFLICTRESOLVER: CONFLICT DETECTION AND WEIGHTED EVIDENCE ARBITRATION

In open web search, different retrieved webpages may contain conflicting claims about the same target fact. If such conflicting evidence is incorporated without explicit resolution, erroneous claims may enter the reasoning context. Once incorporated, these claims may be reused in subsequent processing stages and propagate further through the workflow (Xie et al., 2026). To mitigate this risk, VeriSearch applies ConflictResolver as the final evidence-processing stage before sub-answer generation. Given the evidence set E_t, ConflictResolver detects cross-source factual conflicts and resolves them through weighted evidence arbitration. It uses estimated source authority and webpage recency as heuristic arbitration signals. The resulting evidence set E_t' is then used to generate the sub-answer a_t.

Cross-Source Factual Conflict Detection. ConflictResolver first identifies the target facts required to answer q_t. It then examines whether different webpages provide incompatible claims about the same target fact. The detection prompt applies explicit conflict-detection criteria to exclude spurious cases. A claim pair is treated as a cross-source factual conflict only if it would lead to different answers to q_t. Disagreements on facts unrelated to q_t, differences in detail level or emphasis, and stylistic or phrasing differences do not trigger arbitration. For each detected conflict, the LLM records the conflicting attribute, the specific claims, and their source webpages. If no conflict is detected, the evidence is passed to E_t' without arbitration.

Weighted Evidence Arbitration. When a cross-source factual conflict is detected, ConflictResolver applies weighted evidence arbitration. Because external ground truth is unavailable at inference time, we use estimated source authority and webpage recency as two heuristic signals for evidence arbitration. For each evidence item involved in a detected conflict, it computes an authority score, a recency score, and an arbitration score.

(1) Authority score: For the source URL u_j of each conflicting evidence item, the LLM estimates source authority based on the domain type and publishing entity. The authority score is defined as:

$$\mathrm{Auth}(e_j) = \mathrm{LLM}_{\mathrm{auth}}(u_j) \in [0, 1], \tag{3}$$

where $\mathrm{LLM}_{\mathrm{auth}}$ denotes the task-specific evaluator for source authority.

(2) Recency score: For each detected conflict, ConflictResolver computes a recency score for every conflicting evidence item. The more recent of the two last update dates is used as the reference baseline τ_{ref}. The recency score is defined as:

$$\mathrm{Rec}(e_j) = \max(0.5,\ 1.0 - 0.01 \times \Delta(e_j, \tau_{\mathrm{ref}})), \tag{4}$$

where $\Delta(e_j, \tau_{\mathrm{ref}})$ denotes the day difference between the last update date τ_j of e_j and τ_{ref}. The most recently updated webpage receives a score of 1.0. Older evidence items receive a linearly lower score of 0.01 per day, with a floor of 0.5 to limit the influence of age alone.

(3) Arbitration score: The arbitration score is computed as

$$\Phi(e_j) = \alpha \cdot \mathrm{Auth}(e_j) + (1 - \alpha) \cdot \mathrm{Rec}(e_j), \tag{5}$$

where $\alpha \in (0, 1)$ controls the relative weights of source authority and webpage recency. We set $\alpha = 0.6$ in our experiments, giving slightly greater weight to source authority. This fixed weighted combination defines a simple heuristic arbitration rule. The evidence item with the higher arbitration score is preferred. When the scores tie, the framework uses the ranking fallback path described below.

Conflict Resolution and Sub-Answer Generation. ConflictResolver applies one of two resolution paths according to the arbitration scores.

(1) Score-based path: When the arbitration scores differ, the lower-scoring evidence item is excluded from E'_t, while the higher-scoring item is retained.

(2) Ranking fallback path: When the arbitration scores are equal, the search-engine ranks of the corresponding original search results serve as the final tie-breaker. Both evidence items remain in E'_t. The higher-ranked item is marked as higher-priority evidence, while the lower-ranked item is marked as lower-priority evidence. This preserves potentially useful evidence while indicating which item should be prioritized during sub-answer generation.

The claim supported by the preferred evidence is recorded as the arbitrated claim $\mathcal{F}$. After all detected conflicts are resolved through the applicable path, ConflictResolver generates the sub-answer a_t from the resulting evidence set E'_t and passes it to the Planner for sufficiency assessment and final answer aggregation.

3 Experiments

3.1 Experimental Setup

3.1.1 Datasets

We evaluate VeriSearch on three public benchmarks: xbench-DeepSearch (Xbench-Team, 2025), SEAL-0 (Pham et al., 2026), and Bamboogle (Press et al., 2023). xbench-DeepSearch is a Chinese agent-level benchmark with 100 samples and evaluates end-to-end tool use and multi-step reasoning capabilities. SEAL-0 contains 111 factual questions and evaluates performance under retrieval results that may be conflicting, noisy, or uninformative. Bamboogle contains 125 manually crafted queries designed to assess multi-hop reasoning and knowledge composition.

3.1.2 Baselines

We compare VeriSearch with three categories of baselines: direct reasoning models without retrieval, closed-source agents, and open-source agents. The complete model list and corresponding citations are provided in Appendix A.

For same-backbone comparisons, we reimplement WebThinker-Base (Li et al., 2026) and Search-o1 (Li et al., 2025) with Qwen3.5-9B. MindSearch (Chen et al., 2025) adopts a "code as planning" paradigm with a protocol-specific output format. In our reimplementation of MindSearch using Qwen3.5-9B, we observed that the model did not reliably adhere to MindSearch's output protocol, as detailed in Appendix E. We therefore evaluate MindSearch with its officially supported InternLM2.5-7B-Chat backbone.

3.1.3 Evaluation Metrics

Following Team et al. (2025b), we adopt LLM-as-a-Judge to evaluate answer correctness: GPT-4o compares each generated answer with the ground truth and determines whether it is correct. We report model-based evaluation (MBE) for the main results and ablations. For the conflict-triggered subset in Section 3.4, we additionally report token-level F1 to measure token overlap between the generated and ground-truth answers. To examine whether the main performance trends depend on the model-based evaluator, we additionally report Exact Match (EM) and token-level F1 for the main benchmark comparisons in Appendix D.

3.1.4 Implementation Details

We use Qwen3.5-9B as the backbone LLM. As shown in the Direct Reasoning block of Table 1, among the evaluated 7B–9B open-source models, Qwen3.5-9B achieves the highest scores on xbench-DeepSearch and Bamboogle. We further evaluate VeriSearch with alternative backbone LLMs in Appendix C, where Qwen3.5-9B achieves the highest performance among the tested backbones on both xbench-DeepSearch and SEAL-0.

Table 1: Comparison of VeriSearch with baselines on xbench-DeepSearch, SEAL-0, and Bamboogle. Results marked with † are obtained from our own evaluation, while the remaining reported results are taken from prior work. A "-" indicates that the corresponding result was not reported in the original work. All scores are percentages. **Bold** and underlined values denote the best and second-best results in each column, respectively.

Model	xbench-DeepSearch	SEAL-0	Bamboogle
Direct Reasoning (w/o Retrieval)			
o4-mini	22.3	2.7	79.2†
DeepSeek-R1-0528	37.0	5.4	73.6†
Qwen3.5-9B	13.0†	0.9†	56.8†
GLM-Z1-9B-0414	12.0†	3.6†	40.8†
Hunyuan-7B-Instruct	6.0†	2.7†	10.4†
Llama-3.1-8B-Instruct	5.0†	3.6†	48.0†
InternLM2.5-7B-Chat	2.0†	1.8†	32.0†
Closed-Source Agents			
Grok-DeepResearch	50.0	-	-
Claude-4-Sonnet	64.6	<u>37.8</u>	-
Doubao-DeepThink	50.0	-	-
OpenAI-o3	<u>66.7</u>	17.1	-
Open-Source Agents			
Kimi-K2-0905	61.0	25.2	-
GLM-4.5-Air-106B	65.0	30.6	-
MindSearch (InternLM2.5-7B-Chat)	50.0†	6.3†	73.6†
WebThinker-Base (Qwen3.5-9B)	42.0†	29.7†	85.6†
Search-o1 (Qwen3.5-9B)	52.0†	30.6†	<u>88.0</u>†
VeriSearch (Ours, Qwen3.5-9B)	**67.3 ± 0.6**	**45.1 ± 0.9**	**89.1 ± 1.9**

For web retrieval, we query Google Search through the Serper API (Serper, 2025) and retrieve the top 6 results for each query. We set the AnswerabilityPruner answerability threshold to $\theta = 4$, the maximum number of reflective retries to $K = 3$, and the ConflictResolver authority weight to $\alpha = 0.6$. Additional inference configurations are provided in Appendix B.

3.2 Main Results

Table 1 reports the results on the three benchmarks. VeriSearch achieves the highest available score on each benchmark.

Comparison with Larger and Proprietary Models. Despite using a 9B backbone, VeriSearch outperforms several systems built on substantially larger or proprietary models. On xbench-DeepSearch, VeriSearch achieves 67.3%, exceeding OpenAI-o3 at 66.7% by 0.6 points and GLM-4.5-Air-106B at 65.0% by 2.3 points. On SEAL-0, VeriSearch reaches 45.1%, exceeding Claude-4-Sonnet at 37.8% by 7.3 points and GLM-4.5-Air-106B at 30.6% by 14.5 points. These results show that VeriSearch can achieve stronger performance with a 9B backbone than systems built on substantially larger or proprietary models.

Same-backbone Comparison. In the Open-Source Agents block of Table 1, WebThinker-Base and Search-o1 use the same Qwen3.5-9B backbone as VeriSearch. On xbench-DeepSearch, VeriSearch achieves 67.3%, exceeding the stronger same-backbone baseline, Search-o1 at 52.0%, by 15.3 points. On SEAL-0, VeriSearch reaches 45.1%, compared with 30.6% for Search-o1, a gain of 14.5 points. On Bamboogle, VeriSearch improves over Search-o1 from 88.0% to 89.1%, a gain

Table 2: Ablation results on xbench-DeepSearch and SEAL-0. *w/o AnswerabilityPruner* denotes the variant that removes answerability scoring, threshold gating ($\theta = 4$), and the reflective retry mechanism; in this variant, the LLM directly outputs the page IDs to be crawled, while the full crawling and cleaning pipeline is retained.

Configuration	xbench-DeepSearch	SEAL-0
VeriSearch (Full)	**67.3** $\pm$ 0.6	**45.1** $\pm$ 0.9
w/o AnswerabilityPruner	62.5 $\pm$ 0.7	38.7 $\pm$ 1.8
w/o ConflictResolver	66.3 $\pm$ 1.5	39.6 $\pm$ 0.9

Table 3: Results on the conflict-triggered subset of SEAL-0. This subset contains 48 questions where cross-source factual conflicts are detected during the search process. All values are percentages.

Configuration	MBE	F1
VeriSearch (Full)	**56.3**	**49.4**
w/o ConflictResolver	47.2	43.4
Absolute Gain	**+9.1**	**+6.0**

of 1.1 points. These same-backbone gains suggest that the improvements stem from the framework design rather than backbone differences.

3.3 Ablation Experiments

As shown in Table 2, removing AnswerabilityPruner decreases performance by 4.8 points on xbench-DeepSearch and 6.4 points on SEAL-0. This consistent degradation shows that AnswerabilityPruner contributes across both benchmarks. Filtering low-answerability webpages can be useful for search agents because retrieval from the open web may introduce noisy or insufficiently informative results.

Removing ConflictResolver decreases performance by 1.0 point on xbench-DeepSearch and 5.5 points on SEAL-0. This difference is consistent with the benchmark characteristics: xbench-DeepSearch focuses on tool-use capabilities in search and information retrieval scenarios, whereas SEAL-0 includes retrieval scenarios with conflicting evidence. The larger degradation on SEAL-0 suggests that ConflictResolver is beneficial when cross-source factual conflicts are more frequent.

Overall, AnswerabilityPruner provides consistent gains across both benchmarks, while the contribution of ConflictResolver is more pronounced on SEAL-0, where conflicting evidence is part of the benchmark setting.

3.4 Conflict-Triggered Subset Analysis

To more directly evaluate ConflictResolver on questions involving cross-source factual conflicts, we conduct a focused analysis on SEAL-0. Cross-source factual conflicts are detected in 48 of the 111 questions, corresponding to 43.2% of the benchmark. We use these 48 questions to form a fixed conflict-triggered subset and evaluate VeriSearch with and without ConflictResolver while keeping all other settings unchanged.

Table 3 reports the results on this subset. Removing ConflictResolver decreases MBE and F1 by 9.1 and 6.0 percentage points, respectively, demonstrating its contribution to handling cross-source factual conflicts. Combined with the ablation results on the full SEAL-0 benchmark, where removing ConflictResolver causes a 5.5-point drop in MBE, the larger degradation on the conflict-triggered subset further indicates that ConflictResolver is particularly beneficial when cross-source factual conflicts are present.

4 Related Work

4.1 LLM-Based Search Agents

Recent work on search agents has explored two representative directions. Prompt-driven approaches coordinate multiple specialized modules to handle complex search scenarios. MindSearch (Chen et al., 2025) adopts a directed acyclic graph (DAG) decomposition strategy with WebPlanner and WebSearcher modules to decompose complex information-seeking tasks and coordinate multiple web searches. ManuSearch (Huang et al., 2025) decouples planning, searching, and webpage reading into specialized submodules. HiRA (Jin et al., 2025b) separates high-level strategic planning from domain-specific execution through a meta-reasoning planner, adaptive coordinator, and multiple domain expert actors.

Training-based approaches learn search and reasoning behaviors through post-training. DeepResearcher (Zheng et al., 2025) applies end-to-end reinforcement learning in real-world web environments. WebDancer (Wu et al., 2026) combines browsing data construction, trajectory sampling, supervised fine-tuning, and reinforcement learning to improve information-seeking ability.

These approaches primarily improve how search agents plan, execute, or learn search and reasoning behaviors. In contrast, VeriSearch focuses on controlling the quality of retrieved evidence before it enters the reasoning context.

4.2 Retrieval Filtering and Conflict Resolution

Retrieval filtering and evidence refinement have been explored at different stages of the search pipeline. ReST meets ReAct (Aksitov et al., 2023) introduces the SelectLink module for snippet-based relevance filtering and grounded summarization over search-result titles and snippets, without full-page crawling or explicit cross-source factual conflict resolution. Search-o1 (Li et al., 2025) introduces a Reason-in-Documents module that refines retrieved documents before incorporating them into the reasoning chain. WebFilter (Dai et al., 2026) learns to use advanced search operators for source-restricted and time-constrained query formulation. AnswerabilityPruner instead estimates from each search-result snippet whether its corresponding full webpage is likely to answer the current sub-question, and gates full-page crawling accordingly.

Conflict resolution has also been studied for retrieved evidence. TruthfulRAG (Liu et al., 2026) and TCR (Ye et al., 2026) address conflicts between retrieved evidence and parametric knowledge using knowledge graphs and dual contrastive encoders with self-answerability signals, respectively. Ge et al. (2025) incorporate media-source background information to help RAG models handle conflicting evidence in fact checking. EvoTrustRAG (Nie et al., 2026) attributes conflicts among retrieved facts to knowledge evolution, intervention, or unresolved conflict, and conditions subsequent evidence handling on this attribution. In VeriSearch, ConflictResolver resolves cross-source factual conflicts through weighted evidence arbitration using estimated source authority and webpage recency as heuristic arbitration signals.

5 Conclusion

This work investigated explicit evidence control for LLM-based search agents through VeriSearch. AnswerabilityPruner performs answerability-based filtering before full-page crawling, while ConflictResolver resolves cross-source factual conflicts through weighted evidence arbitration before sub-answer generation. Across all three benchmarks, VeriSearch outperforms the evaluated same-backbone agents. On xbench-DeepSearch and SEAL-0, it also exceeds the reported scores of Kimi-K2-0905, GLM-4.5-Air-106B, and the closed-source baselines included in our comparison. The current framework focuses on textual web evidence. Future work will extend VeriSearch to richer evidence types, including multimodal web content.

AI use statement

In this work, we used generative AI tools to aid and polish the writing of the manuscript, including improving grammar, phrasing, clarity, and readability of text written by the authors. We did not

use generative AI tools for retrieval or literature discovery, research ideation or execution, drafting sections of the paper, generating synthetic datasets, or proving mathematical claims. All AI-assisted edits were reviewed by the authors to ensure that they preserved the intended meaning and did not introduce unsupported technical claims, results, or citations. We take responsibility for the final content of this work, including all text and claims.

Reproducibility statement

To support reproducibility, Section 3.1 describes the datasets, baselines, evaluation metrics, and core implementation settings used in our experiments. Appendix A lists the evaluated baselines and corresponding model choices, while Appendix B provides additional inference configurations. The source code is included in the supplementary material and will be made publicly available upon publication.

References

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. Rest meets react: Self-improvement for multi-step reasoning llm agent. *arXiv preprint arXiv:2312.10003*, 2023.

Anthropic. Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

ByteDance Doubao. Doubao, 2025. URL http://www.doubao.com/.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Jiangning Liu, Wenwei Zhang, Kai Chen, and Feng Zhao. Mindsearch: Mimicking human minds elicits deep AI searcher. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xgtXkyqw1f.

Yuqin Dai, Shuo Yang, Guoqing Wang, Yong Deng, Zhanwei Zhang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Changhua Meng, Can Yi, et al. Careful queries, credible results: Teaching rag models advanced web search tools with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30458–30466, 2026.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

Ziyu Ge, Yuhao Wu, Daniel Wai Kit Chin, Roy Ka-Wei Lee, and Rui Cao. Resolving conflicting evidence in automated fact-checking: A study on retrieval-augmented llms. *arXiv preprint arXiv:2505.17762*, 2025.

Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gaole He, Gianluca Demartini, and Ujwal Gadiraju. Plan-then-execute: An empirical study of user trust and team performance when using llm agents as a daily assistant. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2025.

Lisheng Huang, Yichen Liu, Jinhao Jiang, Rongxiang Zhang, Jiahao Yan, Junyi Li, and Xin Zhao. Manusearch: Democratizing deep search in large language models with a transparent and open multi-agent framework. In *EMNLP (Findings)*, pp. 2403–2417, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025a.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. Hira: A hierarchical reasoning framework for decoupled planning and execution in deep search. *arXiv preprint arXiv:2507.02652*, 2025b.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pp. 15696–15707. PMLR, 2023.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *Advances in Neural Information Processing Systems*, 38:120091–120131, 2026.

Shuyi Liu, Yu-Ming Shang, and Xi Zhang. Truthfulrag: Resolving factual-level conflicts in retrieval-augmented generation with knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32168–32176, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Lang Mei, Zhihan Yang, Xiaohan Yu, Huanyao Zhang, and Chong Chen. Ai-searchplanner: Modular agentic search via pareto-optimal multi-objective reinforcement learning. *arXiv preprint arXiv:2508.20368*, 2025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Xi Nie, Hongwei Li, Shenghao Wu, Wenshu Fan, Qiyang Song, and Wenbo Jiang. Evotrustrag: Evolution-aware conflict attribution and evidence handling for reliable retrieval-augmented generation. *arXiv preprint arXiv:2608.07933*, 2026.

OpenAI. Introducing openai o3 and o4-mini, 2025. URL `https://openai.com/zh-Hans-CN/index/introducing-o3-and-o4-mini/`.

Thinh Pham, Nguyen Phan Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. SealQA: Raising the bar for reasoning in search-augmented language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=zWb7ueH16c`.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Zhipeng Qian, Zihan Liang, Yufei Ma, Ben Chen, Huangyu Dai, Jiayi Ji, Chenyi Lei, Wenwu Ou, Xiaoshuai Sun, and Qibin Hou. Plan before search: Search agents need plan. *arXiv preprint arXiv:2605.28354*, 2026.

Tianrui Qin, Qianben Chen, Sinuo Wang, He Xing, King Zhu, He Zhu, Dingfeng Shi, Xinxin Liu, Ge Zhang, Jiaheng Liu, Xitong Gao, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Flash-searcher: Fast and effective web agents via DAG-based parallel execution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QuaJ6kJaBm.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Serper. Serper: Google search api, 2025. URL https://serper.dev.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025b.

Tencent Hunyuan. Hunyuan-7b-instruct, 2025. URL https://huggingface.co/tencent/Hunyuan-7B-Instruct.

Zehua Wang, Zhaojin Zhang, Boyu Qiu, Xiaolong Weng, Ying Xiong, Buzhou Tang, and Min Zhang. Sr-rag: Verifiable multi-hop reasoning via on-the-fly symbolic graph construction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38588–38606, 2026.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, et al. Webdancer: Towards autonomous information seeking agency. *Advances in Neural Information Processing Systems*, 38:120957–120985, 2026.

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

Xbench-Team. Xbench-deepsearch, 2025. URL https://xbench.org/agi/aisearch.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. From spark to fire: Modeling and mitigating error cascades in llm-based multi-agent collaboration. *arXiv preprint arXiv:2603.04474*, 2026.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, 2024.

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

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34423–34431, 2026.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025.

Table 4: Performance of VeriSearch with different backbone LLMs on xbench-DeepSearch and SEAL-0. All scores are MBE percentages. Qwen3.5-9B is the backbone used in the main experiments.

Backbone LLM	**xbench-DeepSearch**	**SEAL-0**
Qwen3.5-9B	67.3	45.1
GLM-Z1-9B-0414	48.0	16.2
Hunyuan-7B-Instruct	51.2	14.4
Llama-3.1-8B-Instruct	36.4	7.2

A Detailed Baseline Configurations

This section provides the complete list of baselines evaluated in our experiments. We categorize them by their architectural paradigms.

(1) Direct reasoning without retrieval: We evaluate o4-mini (OpenAI, 2025), DeepSeek-R1-0528 (Guo et al., 2025), Qwen3.5-9B (Qwen Team, 2026), GLM-Z1-9B-0414 (Glm et al., 2024), Hunyuan-7B-Instruct (Tencent Hunyuan, 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and InternLM2.5-7B-Chat (Cai et al., 2024).

(2) Closed-source agents: We compare against Grok-DeepResearch (x.ai, 2025), Claude-4-Sonnet (Anthropic, 2025), Doubao-DeepThink (ByteDance Doubao, 2025), and OpenAI-o3 (OpenAI, 2025).

(3) Open-source agents: The evaluated systems include Kimi-K2-0905 (Team et al., 2025a), GLM-4.5-Air-106B (Zeng et al., 2025), MindSearch (Chen et al., 2025), WebThinker-Base (Li et al., 2026), and Search-o1 (Li et al., 2025).

B Additional Implementation Details

To accommodate long-document reasoning, the maximum context length of Qwen3.5-9B is set to 65,536 tokens. During inference, we set the temperature to 0.0 and the repetition penalty to 1.1 to ensure deterministic generation. The Planner performs at most $T_{\max} = 10$ planning iterations. The Searcher generates $L = 3$ queries per search attempt.

C Performance Across Backbone LLMs

To assess whether VeriSearch is specific to the Qwen3.5-9B backbone used in the main experiments, we additionally evaluate the framework with GLM-Z1-9B-0414, Hunyuan-7B-Instruct, and Llama-3.1-8B-Instruct. Table 4 reports the results on xbench-DeepSearch and SEAL-0.

VeriSearch achieves 48.0% and 16.2% on xbench-DeepSearch and SEAL-0 with GLM-Z1-9B-0414, respectively. With Hunyuan-7B-Instruct, it obtains 51.2% and 14.4%. With Llama-3.1-8B-Instruct, VeriSearch achieves 36.4% on xbench-DeepSearch and 7.2% on SEAL-0. These results indicate that VeriSearch can be instantiated with multiple backbone LLMs, although its absolute performance varies across backbone choices.

D Additional Evaluation Metrics

The main experiments use model-based evaluation (MBE) to assess answer correctness. To complement MBE, we additionally report Exact Match (EM) and token-level F1, which measure lexical agreement with the reference answers without relying on an LLM-based evaluator. As shown in Table 5, the relative performance under EM and F1 is largely consistent with that observed under MBE, providing complementary evidence that the overall performance trends are not specific to the model-based evaluator.

Table 5: Additional evaluation results using model-based evaluation (MBE), Exact Match (EM), and token-level F1. All values are percentages. WebThinker-Base and Search-o1 use the same Qwen3.5-9B backbone as VeriSearch. For VeriSearch, we report the mean and standard deviation over three runs.

Benchmark	Model	MBE	EM	F1
xbench-DeepSearch	WebThinker-Base	42.0	28.0	45.8
	Search-o1	52.0	29.0	42.3
	VeriSearch	**67.3 $\pm$ 0.6**	**37.3 $\pm$ 2.1**	**59.5 $\pm$ 1.4**
SEAL-0	WebThinker-Base	29.7	21.6	25.2
	Search-o1	30.6	21.6	26.0
	VeriSearch	**45.1 $\pm$ 0.9**	**35.1 $\pm$ 0.9**	**38.4 $\pm$ 0.7**
Bamboogle	WebThinker-Base	85.6	66.4	76.8
	Search-o1	88.0	67.2	79.2
	VeriSearch	**89.1 $\pm$ 1.9**	**70.4 $\pm$ 0.0**	**83.1 $\pm$ 1.4**

Table 6: Performance of MindSearch with InternLM2.5-7B-Chat and our Qwen3.5-9B adaptation on xbench-DeepSearch and Bamboogle. The Qwen3.5-9B results are reported to document the adaptation and are not treated as a faithful same-backbone MindSearch baseline.

Backbone	xbench-DeepSearch	Bamboogle
InternLM2.5-7B-Chat	50.0	73.6
Qwen3.5-9B	1.0	38.4

E MindSearch Adaptation to Qwen3.5-9B

As noted in Section 3.1.2, the original implementation of MindSearch uses InternLM2.5-7B-Chat. To examine whether MindSearch can serve as a same-backbone baseline, we additionally adapted it to Qwen3.5-9B. This section describes the adaptation and explains why we do not treat this configuration as a faithful same-backbone implementation of MindSearch.

MindSearch's WebPlanner represents multi-step information seeking as a directed acyclic graph (DAG), where each node corresponds to a web search (Chen et al., 2025). The planner constructs the graph by generating Python code, an approach termed "code as planning" in the original paper (Chen et al., 2025). This design provides a structured mechanism for decomposing complex information-seeking tasks and coordinating multiple web searches. Its official implementation follows a fixed interaction protocol. At each turn, the backbone must emit a Python code block wrapped in InternLM-style action delimiters. The generated code may call only the predefined `WebSearchGraph` APIs, including `add_root_node`, `add_node`, `add_response_node`, `add_edge`, and `node`. Graph construction is incremental. After receiving the current search results, the model must decide whether to expand the graph or terminate planning by adding a response node.

To adapt this protocol to Qwen3.5-9B, we provided few-shot examples of valid graph-construction code and restated the required output format in the system prompt. We also applied automatic repairs for missing action delimiters and malformed Python outputs when possible. Under this adaptation, we observed compatibility issues between Qwen3.5-9B and the original execution protocol. For example, the model sometimes mixed free-form text with executable code or invoked `WebSearchGraph` with arguments inconsistent with the current graph state, such as referencing nonexistent nodes in `add_edge`. In some cases, it also invoked `add_response_node` after constructing only one or two search nodes, terminating the incremental planning process earlier than intended.

Table 6 compares the two backbone configurations on xbench-DeepSearch and Bamboogle. MindSearch with InternLM2.5-7B-Chat obtains scores of 50.0 and 73.6, respectively, while our Qwen3.5-9B adaptation obtains 1.0 and 38.4. These results indicate that simply replacing the original backbone does not yield a faithful reproduction of MindSearch under our adaptation setting.

These differences should not be interpreted as evidence that the MindSearch framework itself performs poorly with Qwen3.5-9B. The adaptation changes the backbone while retaining an interaction protocol designed around the original implementation. Moreover, the observed execution failures prevent a controlled attribution of the performance differences to the backbone alone. The results instead indicate that our Qwen3.5-9B adaptation does not reliably reproduce the execution behavior of the original InternLM2.5-7B-Chat configuration.

We therefore report the Qwen3.5-9B results only to document this adaptation and do not use them as a same-backbone MindSearch baseline. In the same-backbone comparison with VeriSearch, we include WebThinker-Base and Search-o1, both of which use the same Qwen3.5-9B backbone without requiring the MindSearch-specific interaction protocol.