IRIS: CLIMBING TO THE SEARCH FRONTIER

Anonymous authors
Paper under double-blind review

ABSTRACT

We present Iris-mini and Iris-pro, two search agents trained at the 35B-A3B and 397B-A17B scales, together with the data pipeline and training recipe behind them. Tasks are reverse-constructed from the hyperlink structure of a web corpus: we author multi-hop chains over an entity graph distilled from a seed page and its out-links, rewrite every non-answer entity into a descriptive reference, and admit only questions that a reference model fails closed-book yet solves once the supporting evidence is supplied. These questions are then turned into trajectories, which are filtered at the trajectory- and turn-level before supervised fine-tuning (SFT). The policy is then optimized by reinforcement learning (RL) against live search. We alternate the two stages in a procedure we call *SFT–RL climbing*, returning the hardest solved and most efficient rollouts of each RL round to the next supervised pass. Because inference-time context management is worth more on these benchmarks than most reported differences between systems, we evaluate every benchmark both with and without it, holding the tool set, the context limit, and the judge fixed. All results come from a single ReAct agent. With management enabled, on BrowseComp, BrowseComp-ZH, DeepSearchQA, and Humanity's Last Exam the two models reach $82.2/84.8/86.9/52.3$ and $88.6/85.1/92.9/56.4$, the strongest overall results among open-source search agents in their respective parameter ranges. We plan to release the model weights together with the complete recipe for data construction, training, and evaluation.

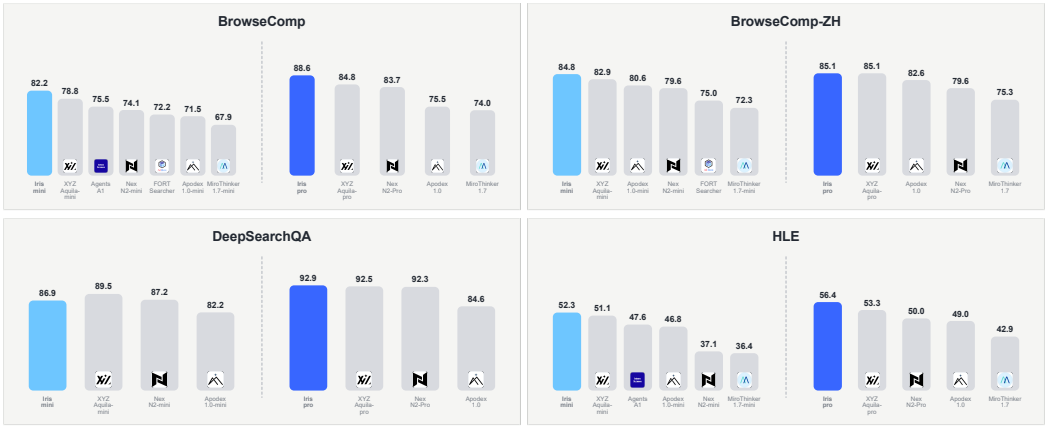

Figure 1: Performance comparison across four agentic search benchmarks.

1 INTRODUCTION

Search agents extend language models beyond closed-form reasoning by allowing them to interact with external tools and retrieve information from dynamic environments. A capable search agent must not only reason about the question, but also decide what to search, how to interpret retrieved evidence, when to continue exploring, and when the available evidence is sufficient to answer. This makes search a fundamentally different setting from conventional language-model evaluation, where

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.

the task, context, and computation budget are largely fixed in advance. Building reliable search agents therefore requires not only stronger models, but also effective training data, long-horizon interaction strategies, and evaluation protocols.

Recent work has made rapid progress toward this goal. Early systems introduced tool use and interleaved reasoning and acting through supervised or self-supervised training (Nakano et al., 2021; Schick et al., 2023; Yao et al., 2023), followed by reinforcement learning with retrieval from static corpora and, more recently, the live web (Li et al., 2025b; Jin et al., 2025; Song et al., 2025; Zheng et al., 2025). Because naturally occurring web questions are often too easy for training, several studies construct harder tasks by traversing hyperlink or knowledge graphs and masking entities along the resulting paths (Wu et al., 2025a; Li et al., 2025a; Tao et al., 2025; Lu et al., 2025). Recent systems have further combined synthetic question generation, trajectory supervision, reinforcement learning, and long-context inference into increasingly standardized pipelines (Tongyi DeepResearch Team, 2025; Du et al., 2026b; MiroMind Team, 2026; Chu et al., 2026; Deng et al., 2026; Kimi Team, 2026; DeepSeek-AI, 2025; Apodex Team, 2026; XYZ Agentic Team, 2026; Bai et al., 2026; Nex-AGI Team, 2025). Despite these advances, substantially different performance can still arise from differences in the inference-time harness rather than from differences in the underlying policy.

A particularly important factor is *context management* (CM). Long-horizon search can exhaust the available context before the agent has resolved all required constraints, making the effective search budget much smaller than the nominal context window. Existing systems address this through trajectory summarization, state compression, selective history removal, or full context reset (Wu et al., 2025b; Ye et al., 2025; DeepSeek-AI, 2025). The impact can be substantial when search trajectories are long, while the benefit diminishes as the available context grows or the agent requires fewer interaction steps (Kimi Team, 2026; Deng et al., 2026; Chu et al., 2026). This suggests that CM should be viewed not simply as an implementation detail, but as part of the effective inference system. Reporting only results under a managed context can therefore obscure how much performance comes from the policy itself and how much comes from the surrounding harness.

In this report, we present an end-to-end recipe for the data construction, training, and evaluation of strong search agents. Our training pipeline starts from automatically constructed multi-hop tasks derived from web-graph structure. We remove easily searchable anchors through entity rewriting and retain only questions that are both sufficiently difficult and objectively verifiable (Section 2). We then collect trajectories from a strong teacher and apply multi-stage filtering at both the trajectory and turn levels to improve the quality of supervised training data. The resulting policy is optimized with reinforcement learning against live search, using an in-cluster judge and observation summarizer, and SFT and RL are alternated iteratively so that successful behaviors discovered during each rollout stage are reinforced before the next round of exploration (Section 3).

We evaluate the resulting agents under two inference regimes: with and without CM. All other major components, including the tool interface, context budget, and judging procedure, are held fixed to make the effect of CM directly measurable. As shown in Figure 1, our Iris-mini and Iris-pro systems achieve the best overall performance among open-source search agents across four challenging benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), DeepSearchQA (Gupta et al., 2026), and Humanity's Last Exam (HLE) (Center for AI Safety et al., 2026).

Our main contributions are summarized as follows:

- We develop an end-to-end pipeline for constructing and verifying challenging multi-hop search tasks from web structure, together with trajectory-level and turn-level filtering to obtain high-quality training data.
- We combine filtered SFT, RL with live web search, and iterative SFT–RL climbing, allowing high-quality trajectories discovered during RL to be fed back into subsequent supervised training rounds.
- We plan to release the model weights, and key components of the data construction, training, and evaluation pipelines to facilitate reproduction and further research on search agents.

2 DATA PIPELINE

A capable search agent can be trained on questions that (i) cannot be answered from parametric memory alone and (ii) require composing evidence dispersed across several sources. Naturally occurring web questions rarely satisfy both conditions, and hand-written questions are expensive and hard to scale. We therefore build a fully LLM-driven data pipeline that reverse-constructs questions from the hyperlink structure of a web corpus. The pipeline is organized into three stages: web-graph construction, task synthesis, and dual-criteria verification.

2.1 WEB-GRAPH CONSTRUCTION

We model the corpus as a directed graph whose nodes are pages and whose edges are hyperlinks,

$$G = (V, E), \qquad \mathrm{Out}(v) = \{\, u \in V \mid (v, u) \in E \,\}. \tag{1}$$

Each synthesis instance begins from a seed page drawn by a sampling policy,

$$v_0 \sim P_{\text{seed}}(V), \tag{2}$$

for which we use the answer-anchored mode that fixes a target answer entity and retrieves pages describing it. We then expand the seed along its out-links into a local subgraph

$$G_{\text{sub}} = \big(\{v_0\} \cup N,\ E_{\text{sub}}\big), \qquad N = \{v_i\}_{i=1}^{k} \subseteq \mathrm{Out}(v_0), \tag{3}$$

retaining the full text of each page truncated to a fixed budget. Because the reader services that render pages strip inline anchors, we recover the true out-link set of v_0 from a structured semantic mirror of the corpus (RDF triples) together with rendered page markup, and merge the two sources to maximize link recall.

2.2 TASK SYNTHESIS

Given a subgraph G_{sub}, this stage produces a single obscured multi-hop question through three steps: distilling the pages into an entity graph, authoring a question over that graph, and abstracting away every directly searchable anchor.

Entity-graph extraction. Raw pages carry substantial noise that distracts question generation. We distill G_{sub} into a compact, connected entity graph

$$G_e = (V_e, R_e) = f_{\text{ext}}(G_{\text{sub}}), \qquad |V_e| \le n, \tag{4}$$

where V_e are salient entities and R_e are typed semantic relations that preserve the cross-page link structure of G_{sub}. The extractor keeps only entities and relations that lie on a multi-hop path toward the seed theme, yielding a dense relational skeleton over which questions can be authored precisely.

Multi-hop question generation. Let $y = \mathrm{theme}(v_0)$ denote the seed theme, which we take as the target answer. We generate an initial question together with its reasoning path over the entity graph,

$$(q_0,\, P) = f_{\text{gen}}(G_e,\, y) \quad \text{s.t.} \quad |P| \ge N \ \wedge\ y \notin q_0, \tag{5}$$

where $P = (e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \to y)$ is a path in G_e whose traversal uniquely yields y. The hard constraint $|P| \ge N$ forces the question to depend on at least N coupled relations.

Anchor abstraction. Concrete anchors let an agent bypass the intended reasoning by directly searching a surface string. We remove this shortcut with an abstraction operator $\mathcal{A}$ that rewrites every non-answer entity into a descriptive reference,

$$\mathcal{A}(e): \quad \mathrm{name}(e),\ \mathrm{alias}(e) \notin \mathcal{A}(e) \ \wedge\ \mathcal{A}(e) \text{ uniquely identifies } e, \qquad \forall\, e \in V_e \setminus \{y\}. \tag{6}$$

The initial question is then rewritten into its abstracted form

$$\tilde{q} = f_{\text{abs}}(q_0, \mathcal{A}), \tag{7}$$

substituting each mentioned entity e by $\mathcal{A}(e)$ while preserving the reasoning structure and the answer y. The resulting $\tilde{q}$ demands disambiguation-by-reasoning rather than string matching.

2.3 DUAL-CRITERIA VERIFICATION

We admit a pair $(\tilde{q}, y)$ only if it is simultaneously hard and solvable, judged by a reference model M_{ref} under two settings,

$$c_{\text{diff}}(\tilde{q}) = \mathbb{I}[\, M_{\text{ref}}(\tilde{q}) \neq y \,], \qquad c_{\text{solv}}(\tilde{q}) = \mathbb{I}[\, M_{\text{ref}}(\tilde{q} \mid G_e) = y \,], \tag{8}$$

and we keep only the intersection

$$\mathcal{D} = \big\{ (\tilde{q}, y) \;\big|\; c_{\text{diff}}(\tilde{q}) \cdot c_{\text{solv}}(\tilde{q}) = 1 \big\}. \tag{9}$$

The difficulty criterion (closed-book, no tools) discards simple questions the model already answers from memory; the solvability criterion (with G_e supplied as context) discards questions whose answer is wrong or non-unique. Answer equality $M_{\text{ref}}(\cdot) = y$ is decided by a semantic-matching judge. The accepted set $\mathcal{D}$ of abstracted multi-hop questions, each paired with a verified, unique answer, along with some in-house and open-source question sets (Du et al., 2026b; Zhao et al., 2026; Chu et al., 2026; XYZ Agentic Team, 2026), will be used for trajectory generation.

3 TRAINING RECIPE

We next train the search agent through iterative cycles of SFT and RL.

3.1 SUPERVISED FINE-TUNING

Trajectory generation. We prompt a strong teacher M_T to solve each question $q \in \mathcal{D}$ under the ReAct paradigm (Yao et al., 2023), interleaving reasoning, tool calls, and observations against live search tools. A trajectory is the resulting sequence of reasoning steps, tool calls, and observations, terminated by a final answer,

$$\tau = \big(r_1, a_1, o_1, \ldots, r_T, a_T, o_T, r_{T+1}, \hat{y}\big) \;\sim\; \pi_{M_T}\big(\cdot \mid q, \mathcal{T}\big), \tag{10}$$

where r_t is the reasoning at step t, $a_t \in \mathcal{T}$ is a tool call from the tool set $\mathcal{T} = \{\text{SEARCH}, \text{SCRAPE}\}$, o_t is the returned observation, and $\hat{y}$ is the final answer. Each observation is a document-level summary produced on the fly rather than a raw page, which keeps trajectories within a bounded context budget.

Coarse filtering. Let $\mathcal{D}_{\text{raw}} = \{(q, \tau)\}$ be the pool of collected trajectories. Every sample must clear a trajectory-level stage that admits τ only if it is correct, non-degenerate, and non-trivial. Correctness is a two-part gate: the rollout must terminate successfully, and its answer must be judged correct by an LLM judge J against the reference y^*,

$$c_{\text{corr}}(q, \tau) = \mathbb{I}[\text{success}(\tau)] \cdot \mathbb{I}[J(q, \hat{y}_\tau, y^*) = \text{CORRECT}]\,. \tag{11}$$

We then remove degenerate trajectories: repetition loops, runaway tool calling, unterminated `<think>` blocks, and other pathologies. Our primary detector is a sliding-window compression ratio, computed in $O(n)$ over the decoded text,

$$\rho_{\text{cr}}(w) = \frac{|w|}{\big|\text{zlib}(w)\big|}, \qquad c_{\text{degen}}(\tau) = \mathbb{I}\Big[\max_{w} \rho_{\text{cr}}(w) \geq \tau_{\text{cr}}\Big]\,. \tag{12}$$

Repetitive text compresses far more than fluent text, so any window whose ratio reaches τ_{cr} signals a loop, whatever its period and wherever it begins. Auxiliary detectors cover what compression alone can miss: periodically repeating lines, long single-character runs, and consecutive tool calls with byte-identical arguments. Finally we keep only trajectories with at least K tool-call turns, discarding shallow cases a direct lookup could resolve,

$$c_{\text{depth}}(\tau) = \mathbb{I}[T_{\text{tool}}(\tau) \geq K]\,. \tag{13}$$

An exact-duplicate pass over full message sequences then drops byte-identical trajectories while keeping distinct rollouts of the same question, giving the coarse set

$$\mathcal{D}_{\text{sft}} = \text{dedup}\big\{(q, \tau) \in \mathcal{D}_{\text{raw}} \;\big|\; c_{\text{corr}} \cdot (1 - c_{\text{degen}}) \cdot c_{\text{depth}} = 1\big\}. \tag{14}$$

Fine filtering. Coarse filtering keeps or drops whole trajectories. When a question sits near the capability boundary of the teacher, however, a correct trajectory may still contain a locally poor turn: a redundant search, a hallucinated tool name, or reasoning inconsistent with the action actually taken. As a turn-level refinement, we label individual turns with an LLM judge. The difficulty is that a turn which looks wasteful in isolation is often a legitimate exploratory step, and because no human is in the loop, the judge has to draw that line on its own. We therefore induce the criteria from the data rather than hand-crafting them. Specifically, we sample some trajectories and ask the judge to critique them in free form, then consolidate the recurring failure modes (e.g., misinterpreting the question) into an explicit rubric used in the final judging prompt. For each assistant turn, the judge receives the question, reference answer, and a fixed local window of surrounding turns, and outputs either KEEP or MASK, represented as $m_t \in \{0, 1\}$. To prevent overly aggressive filtering, we mask at most 10% of the assistant turns in any trajectory. Masked turns remain in the context but are excluded from the training loss, providing cleaner learning signals without discarding useful interaction history.

Training objective. For each assistant turn, let $C_{<t}$ denote its visible history, reconstructed by the harness's replay operator from the append-only conversation so as to stay byte-identical to what the agent conditioned on at inference. Over $\mathcal{D}_{\text{sft}}$ we maximize the likelihood of the teacher's output,

$$\mathcal{L}_{\text{SFT}}(\theta) = -\,\mathbb{E}_{(q,\tau)\sim\mathcal{D}_{\text{sft}}} \sum_{t=1}^{T+1} m_t \,\log \pi_\theta\big(u_t \mid C_{<t}\big), \tag{15}$$

where u_t is the assistant output at turn t, namely the reasoning and tool call (r_t, a_t) for $t \le T$ and the final reasoning and answer $(r_{T+1}, \hat{y})$ at $t = T + 1$. The per-turn mask $m_t \in \{0, 1\}$ is supplied by fine filtering; observation, user, and system tokens carry zero loss by construction.

3.2 Reinforcement Learning

We then optimize the agent against live search with a group-relative policy gradient. To keep long-horizon rollouts affordable without going fully asynchronous, we place the rollout regime *between* on- and off-policy through request-level partial rollout. To avoid depending on external APIs at training time, we co-locate an in-house model that serves as *both* reward judge and observation summarizer.

Partial rollout and prefix reuse. Long-horizon rollouts have a heavy tail: a few sessions run far longer than the rest and stall a synchronous step. Rather than discard unfinished work, as task-level streaming does, we interrupt at the request level. Once a step has committed enough completed trajectories, in-flight over-sampled sessions are aborted and resumed at the next step from their committed prefix. Rollouts are organized as a forest whose nodes are message states, each caching its token, loss-mask, log-probability, and weight-version deltas, so a resumed trajectory is a path that splices prefixes generated under different policy weights, a mismatch we correct with truncated importance sampling. Completed turns are therefore reused rather than thrown away, which keeps rollout GPUs busy under a synchronous, co-located schedule at the cost of roughly $2\times$ over-sampling as headroom.

In-house reward and summarization. Rather than call an external API, we run several FP8 engines of an in-house Qwen3.5-397B-A17B model inside the training cluster, alongside the actor and rollout engines. Acting as a generative reward model (GenRM), the model scores a rollout with a binary verdict on the extracted answer $\hat{y}_\tau$ against the reference y^*,

$$R(q, \tau) = \mathbb{I}\big[\mathrm{GenRM}\big(q, \hat{y}_\tau, y^*\big) = \mathrm{A}\big]\,, \tag{16}$$

with no additive format term, since an empty or mid-thought-truncated answer extracts to $\varnothing$ and already scores 0 without invoking the judge. As a summarizer, the same engines compress each retrieved page into a short, query-relevant digest that supplies the observation o_t of Eq. (10). This is the only context-reduction mechanism in the rollout, with no message-history pruning or sliding window, so the context the policy conditions on is exactly what is trained on. Serving both roles in-cluster removes the external API dependency from the training loop, and one allocation covers training, rollout, and reward.

3.3 ITERATIVE CLIMBING

We alternate SFT and RL in iterative cycles. Each round of RL explores the current policy, after which a small set of high-quality rollouts is distilled back into the policy through supervised fine-tuning; RL then resumes from the updated model. We call each such cycle a *climb*. This alternation complements the strengths of the two objectives: RL improves behaviors sampled by the current policy through relative rewards, while supervised fine-tuning directly reinforces rare but successful trajectories instead of relying on their contribution to a group-relative gradient.

For each query q, we retain at most one rollout from the existing RL samples. Let $\bar{R}(q)$ denote the pass rate within its rollout group. We select queries with $0 < \bar{R}(q) \leq 1/2$, targeting cases that are solvable but not yet reliable. Among successful rollouts, we require at least K_{rft} tool-call turns and then select the shortest valid trajectory:

$$\mathcal{S}_q = \big\{\tau_i \mid R(q,\tau_i) = 1,\ T_{\text{tool}}(\tau_i) \geq K_{\text{rft}}\big\}, \qquad \tau_q^\star = \arg\min_{\tau \in \mathcal{S}_q} T_{\text{tool}}(\tau). \tag{17}$$

The depth constraint filters out lucky or trivial solutions, while the shortest-trajectory criterion discourages unnecessary search. The resulting set is deduplicated and used for SFT before the next RL round. Since the difficulty band is defined by the current policy's pass rate, it automatically shifts toward harder examples as the policy improves, providing a simple self-paced curriculum and a natural stopping signal as the candidate pool diminishes. Further details will be released in the future.

4 EVALUATION

4.1 SETUP

Benchmarks. We evaluate our models on four challenging benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), DeepSearchQA (Gupta et al., 2026), and HLE (Center for AI Safety et al., 2026). BrowseComp evaluates an agent's ability to identify long-tail entities and provide concise answers based on multiple indirect and mutually constraining clues. BrowseComp-ZH follows the same evaluation setting but focuses on Chinese-language sources. DeepSearchQA evaluates the comprehensiveness of search-based answers rather than the correctness of a single answer span, with scores reflecting the proportion of required evidence successfully recovered by the agent. HLE evaluates expert-level academic reasoning across a broad range of disciplines, where information retrieval is expected to complement rather than substitute for domain knowledge. We evaluate the text-only subset of HLE.

Implementation Details. We initialize Iris-mini and Iris-pro from Qwen3.6-35B-A3B and Qwen3.5-397B-A17B respectively, both of which adopt a mixture-of-experts (MoE) architecture with a 256K-token context window. For SFT, the models are trained for two epochs with a global batch size of 64 and a maximum sequence length of $262{,}144$ tokens. For RL, we use the open-source Relax framework (Zhang et al., 2026) as the underlying training engine.

To prevent models from exploiting benchmark leakage, we block access to the Hugging Face dataset and Space pages that host the benchmark questions and answers (i.e., `huggingface.co/datasets` and `huggingface.co/spaces`), enforced at three points: removed from search results, refused on scrape, and caught by a post-hoc guard in the tool manager even when the model supplies the URL from memory.

Context Management. We believe that specifically designing different CM strategies for different benchmarks is of little practical significance. Therefore, for all benchmarks, our main results are reported under two settings: without CM, and with the *discard-all* strategy of DeepSeek-V3.2 (DeepSeek-AI, 2025). In Section 4.3, we further compare a broader range of CM strategies and evaluate their performance across different benchmarks.

Evaluation Protocol. We perform a single rollout (pass@1) for each question. The resulting final answer is evaluated against the corresponding reference answer using an LLM-based judge with each benchmark's official evaluation prompt. DeepSearchQA is evaluated using F1, whereas BrowseComp, BrowseComp-ZH, and HLE are evaluated using accuracy. All benchmarks are evaluated under

Table 1: Main results on the four search benchmarks with CM enabled for each system. For our models we report the discard-all setting throughout; the remaining strategies are analyzed in Table 2. HLE uses the text-only setting by default; results evaluated on the full set are marked with ^{f}. ^{r} indicates that the corresponding result was reproduced by the XYZ-Aquila Team.

Model	Size	BrowseComp	BrowseComp-ZH	DeepSearchQA	HLE
30∼35B					
MiroThinker-1.7-mini	30B	67.9	72.3	–	36.4
FORT-Searcher	30B	72.2	75.0	–	–
Agents-A1	35B	75.5	–	–	47.6
Nex-N2-mini	35B	74.1	79.6^{r}	87.2^{r}	37.1^{r}
Apodex-1.0-mini	35B	71.5	80.6	82.2	46.8
XYZ-Aquila-mini	35B	78.8	82.9	**89.5**	51.1
Iris-mini	35B	**82.2**	**84.8**	86.9	**52.3**
∼ 400B					
MiroThinker-1.7	397B	74.0	75.3	–	42.9
Nex-N2-Pro	397B	83.7	79.6^{r}	92.3^{r}	50.0^{r}
Apodex-1.0	397B	75.5	82.6	84.6	49.0
XYZ-Aquila-pro	397B	84.8	**85.1**	92.5	53.3
Iris-pro	397B	**88.6**	**85.1**	**92.9**	**56.4**
Frontier / heavy-compute					
MiroThinker-H1	–	88.2	**84.4**	–	47.7
Apodex-1.0-H	–	90.3	84.1	94.4	60.8
Kimi-K2.6	1T	83.2	–	92.5	54.0^{f}
GLM-5.2	744B	–	–	–	54.7
DeepSeek-V4-Pro (Preview)	1.6T	83.4	–	–	48.2
Claude Fable 5	–	88.0	–	94.2	**64.5**^{f}
GPT-5.6 Sol	–	90.4	–	–	58.0^{f}
Kimi-K3	2.8T	**91.2**	–	**95.0**	56.0^{f}
Qwen3.8-Max	2.4T	–	–	–	56.2^{f}

a consistent configuration, including the same tool set, context-length limit, and maximum turn budget.

4.2 Experimental Results

Table 1 summarizes the main results on the four benchmarks. For our models we report the discard-all setting, which we treat as the default configuration; the remaining strategies are examined in Section 4.3.

In the 30–35B parameter range, Iris-mini achieves the best performance on three benchmarks: BrowseComp (82.2), BrowseComp-ZH (84.8), and HLE (52.3). On BrowseComp, it outperforms the strongest model in the same parameter range, XYZ-Aquila-mini (XYZ Agentic Team, 2026), by 3.4 points (82.2 vs. 78.8). On DeepSearchQA, it achieves an F1 score of 86.9, which remains below XYZ-Aquila-mini (89.5).

In the ∼400B parameter range, Iris-pro leads or matches the best result on all four benchmarks. It achieves the highest score on BrowseComp (88.6), outperforming XYZ-Aquila-pro by 3.8 points, the highest F1 on DeepSearchQA (92.9), and the highest accuracy on HLE (56.4), ahead of XYZ-Aquila-pro by 3.1 points. On BrowseComp-ZH it ties XYZ-Aquila-pro at 85.1.

These results demonstrate that our systems are competitive relative to models with substantially larger parameter counts. Iris-mini approaches the performance of 1T-scale models such as Kimi-K2.6 and DeepSeek-V4-Pro on BrowseComp. This highlights the effectiveness of our training approach in specializing a relatively compact model for web search and long-horizon information-seeking tasks. Iris-pro further demonstrates strong intrinsic search capabilities: even in the standard ReAct setting, it achieves performance comparable to or better than MiroThinker-H1 and Apodex-1.0-H under their heavy-compute configurations. These results reflect substantial improvements in

Table 2: Effect of CM. For each system, w/o denotes the baseline without CM. The subsequent rows show the results of different CM strategies, with the change relative to the corresponding baseline reported in parentheses.

Context management	BrowseComp	BrowseComp-ZH	DeepSearchQA	HLE
OpenSeeker-v2 (30B)				
w/o	46.0	58.1	–	34.6
REDSearcher (30B)				
w/o	42.1	49.8	–	34.3
discard-all	57.4 (+15.3)	58.2 (+8.4)	–	–
FORT-Searcher (30B)				
w/o	55.9	62.1	–	–
discard-all	72.2 (+16.3)	75.0 (+12.9)	–	–
Iris-mini (35B)				
w/o	64.7	72.3	81.0	43.2
retry	–	83.0 (+10.7)	89.1 (+8.1)	52.0 (+8.8)
discard-all	82.2 (+17.5)	84.8 (+12.5)	86.9 (+5.9)	52.3 (+9.1)
discard-all + retry	**85.9** (+21.2)	**85.1** (+12.8)	**89.9** (+8.9)	**52.4** (+9.2)
Iris-pro (397B)				
w/o	72.6	76.8	86.4	50.8
retry	–	84.1 (+7.3)	92.3 (+5.9)	**56.6** (+5.8)
discard-all	88.6 (+16.0)	**85.1** (+8.3)	92.9 (+6.5)	56.4 (+5.6)
discard-all + retry	**90.3** (+17.7)	**85.1** (+8.3)	**93.4** (+7.0)	**56.6** (+5.8)

the underlying search intelligence of our models. Nevertheless, our models still leave a gap relative to the most capable frontier systems. Closing this gap remains an important direction for future work.

4.3 Analysis

Table 2 investigates the effect of CM on our search agents. In addition to *discard-all*, which clears the entire interaction history and restarts the question from scratch when the running context reaches a predefined threshold before a final answer is obtained, we also evaluate the *retry* strategy proposed by MiroThinker (MiroMind Team, 2026). When an attempt fails to produce a valid answer, *retry* summarizes the failed attempt into a compact description of the unsuccessful search experience and appends it to the task for the next attempt. In this sense, retry can be viewed as a form of discard-all with progressively accumulated prior knowledge about what has already been explored and ruled out. We do not consider additional inference-time verification or re-answering procedures adopted by some systems, such as the *reverify* and *reanswer* strategies used by XYZ-Aquila (XYZ Agentic Team, 2026) on BrowseComp-ZH and related benchmarks, as these introduce additional test-time mechanisms beyond CM itself.

Several prior works explicitly report results without any CM, including OpenSeeker-v2 (Du et al., 2026a), REDSearcher (Chu et al., 2026), and FORT-Searcher (Deng et al., 2026). We believe such results are particularly informative because they provide a more direct measurement of the search agent itself. Without an additional CM mechanism, the reported performance is determined primarily by the quality of the training data and the search behavior learned by the model, rather than by how much performance can be recovered through an external inference-time wrapper. We therefore also report these no-CM results in Table 2 for a more direct comparison. This distinction is important because many existing results are reported only with CM enabled, making it difficult to disentangle the intrinsic capability of the agent from the gains introduced by the inference-time strategy.

Our models demonstrate a clear advantage in both regimes. Without any CM, Iris-mini achieves 64.7 on BrowseComp and 72.3 on BrowseComp-ZH, outperforming FORT-Searcher (55.9 and 62.1), OpenSeeker-v2 (46.0 and 58.1), and REDSearcher (42.1 and 49.8). Iris-pro raises these two scores by a further 7.9 and 4.5 points, respectively. More importantly, the advantage is retained after CM is introduced, indicating that our gains are not solely attributable to the inference-time context

strategy. Instead, the underlying search capability learned by our models already provides a strong foundation, which CM can further exploit.

Across the evaluated settings, CM consistently improves performance over the corresponding no-CM baseline, with substantially larger gains observed for Iris-mini than for Iris-pro. What differs is not the size of the budget but how quickly each model consumes it: the smaller model needs more steps to resolve the same set of constraints, so it reaches the limit more often and CM has correspondingly more to recover. CM is therefore most valuable when a model has already learned effective search behaviors but exhausts its context before it can bring them to bear.

The gains are also strongly benchmark-dependent. For Iris-mini, CM is worth up to 21.2 points on BrowseComp, against at most 12.8 on BrowseComp-ZH, 9.2 on HLE, and 8.9 on DeepSearchQA. This ordering is not explained by how much room each benchmark leaves: HLE has the lowest no-context baseline of the four at 43.2, yet it gains less than BrowseComp. What the ordering does follow is how often a session actually runs out of context. BrowseComp-style tasks require long-horizon information seeking, in which the agent repeatedly retrieves, filters, and integrates evidence across many search steps; as the interaction history grows, the context itself becomes the binding constraint, and resetting the accumulated tool history buys additional search on the same question. HLE instead places its emphasis on expert-level knowledge and reasoning, for which web retrieval is complementary rather than the primary source of the answer. Its deficit is therefore not a context problem, and extending the effective search horizon recovers correspondingly less.

Another notable observation is how tightly the two model scales converge on BrowseComp-ZH. Three configurations land on exactly the same score of 85.1: Iris-mini under discard-all with retry, and Iris-pro under both discard-all and discard-all with retry. On a benchmark of 289 questions this corresponds to 246 correct answers in all three cases, suggesting that, at least on this benchmark, the remaining performance is constrained by factors beyond model capacity.

Finally, combining discard-all with retry achieves the strongest result in most settings and pushes Iris-pro above 90 on BrowseComp. However, this improvement comes at a substantial inference cost, since each retry requires another full search attempt. We therefore regard retry as an interesting exploration of the upper bound achievable through increasingly aggressive inference-time strategies, rather than as the primary configuration for reporting model performance. Accordingly, Table 1 reports the discard-all results for all models, even when retry can produce a higher score. This concern is not ours alone: the same point is made in DeepSeek-AI (2025), which argues that actual compute cost has to be accounted for when benchmarking CM strategies. Our goal is not simply to maximize a benchmark score through increasingly elaborate inference-time procedures, but to train a genuinely capable search agent. Benchmark performance captures only part of an agent's overall capability, and we therefore consider evaluations in a broader range of realistic search scenarios to be an important direction beyond these benchmark results.

5 Conclusion: Beyond Search

In this report, we presented an end-to-end recipe for building search agents, covering data construction, training, and evaluation, and instantiated it at two scales: Iris-mini and Iris-pro. Our recipe combines verified tasks synthesized from web graphs, trajectory- and turn-level filtering, and iterative SFT–RL climbing against live search. Across four widely used benchmarks, both models achieve the best overall performance within their respective parameter ranges. These results support the effectiveness of our training recipe and motivate further evaluation across a broader range of agentic scenarios.

One observation from our experiments points beyond search itself. Both the synthesized search data and the search-specialized models, the latter used as teachers for on-policy distillation, transferred positively to domains that were not explicitly targeted, including General Tool Use and Cowork. We interpret this as evidence that search may be better viewed as an atomic capability than as a vertical specialization: the behaviors it induces are not limited to web browsing and appear reusable wherever an agent must act under incomplete information. If this observation generalizes, search data and search-derived teachers may be useful throughout training rather than confined to a separate specialization stage, with their value measured not only by browsing performance but also by broader agentic competence. This is the direction we intend to pursue.

REFERENCES

Apodex Team. Apodex-1.0: A verification-centric agent team for discoverative intelligence. Technical report, 2026. URL https://www.apodex.com/pdf/20260608.

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, Xiaohan He, Shuyue Hu, Yusong Hu, Songtao Huang, Yichen Jiang, Hao Li, Xin Li, Dahua Lin, Weihao Lin, Fenghua Ling, Dongrui Liu, Zhuo Liu, Wenjie Lou, Runmin Ma, Chunjiang Mu, Haoyang Peng, Tianshuo Peng, Jinxin Shi, Luohe Shi, Boyuan Sun, Zelin Tan, Shengji Tang, Yan Teng, Qianyi Wang, Xiaosong Wang, Yiming Wu, Yi Xie, Xiangchao Yan, Jingqi Ye, Peng Ye, Fangchen Yu, Jiakang Yuan, Bihao Zhan, Bo Zhang, Chen Zhang, Shufei Zhang, Shuaiyu Zhang, Wenlong Zhang, Yiqun Zhang, Junpeng Zhao, Zhijie Zhong, Bowen Zhou, and Yuhao Zhou. Scaling the horizon, not the parameters: Reaching trillion-parameter performance with a 35B agent. arXiv preprint arXiv:2606.30616, 2026. URL https://arxiv.org/abs/2606.30616.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://doi.org/10.1038/s41586-025-09962-4.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, Dongdong Kuang, Ming Liu, Bing Qin, and Xing Yu. REDSearcher: A scalable and cost-efficient framework for long-horizon search agents. arXiv preprint arXiv:2602.14234, 2026. URL https://arxiv.org/abs/2602.14234.

DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models. arXiv preprint arXiv:2512.02556, 2025. URL https://arxiv.org/abs/2512.02556.

Jia Deng, Yimeng Chen, Xiaoqing Xiang, Ziyang Zeng, Shuo Tang, Wayne Xin Zhao, Feng Chang, Chuan Hao, Yuan Wei, Ran Tao, Bryan Dai, and Ji-Rong Wen. FORT-Searcher: Synthesizing shortcut-resistant search tasks for training deep search agents. arXiv preprint arXiv:2606.12087, 2026. URL https://arxiv.org/abs/2606.12087.

Yuwen Du, Rui Ye, Shuo Tang, Keduan Huang, Xinyu Zhu, Yuzhu Cai, and Siheng Chen. OpenSeeker-v2: Pushing the limits of search agents with informative and high-difficulty trajectories. arXiv preprint arXiv:2605.04036, 2026a. URL https://arxiv.org/abs/2605.04036.

Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. OpenSeeker: Democratizing frontier search agents by fully open-sourcing training data. arXiv preprint arXiv:2603.15594, 2026b. URL https://arxiv.org/abs/2603.15594.

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

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. arXiv preprint arXiv:2503.09516, 2025. URL https://arxiv.org/abs/2503.09516.

Kimi Team. Kimi K3: Open frontier intelligence. arXiv preprint arXiv:2607.24653, 2026. URL https://arxiv.org/abs/2607.24653.

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

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. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. DeepDive: Advancing deep search agents with knowledge graphs and multi-turn RL. arXiv preprint arXiv:2509.10446, 2025. URL https://arxiv.org/abs/2509.10446.

MiroMind Team. MiroThinker-1.7 & H1: Towards heavy-duty research agents via verification. arXiv preprint arXiv:2603.15726, 2026. URL https://arxiv.org/abs/2603.15726.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

Nex-AGI Team. Nex-N1: Agentic models trained via a unified ecosystem for large-scale environment construction. arXiv preprint arXiv:2512.04987, 2025. URL https://arxiv.org/abs/2512.04987.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning. arXiv preprint arXiv:2503.05592, 2025. URL https://arxiv.org/abs/2503.05592.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebShaper: Agentically data synthesizing via information-seeking formalization. arXiv preprint arXiv:2507.15061, 2025. URL https://arxiv.org/abs/2507.15061.

Tongyi DeepResearch Team. Tongyi DeepResearch technical report. arXiv preprint arXiv:2510.24701, 2025. URL https://arxiv.org/abs/2510.24701.

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

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebDancer: Towards autonomous information seeking agency. In *Advances in Neural Information Processing Systems*, volume 38, pp. 120957–120985. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4035. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/af043aee9cb137785d93195c6cf4cd96-Abstract-Conference.html.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. ReSum: Unlocking long-horizon search intelligence via context summarization. arXiv preprint arXiv:2509.13313, 2025b. URL https://arxiv.org/abs/2509.13313.

XYZ Agentic Team. AI4AI at scale: A full-pipeline system for enhancing LLM agentic capabilities. Technical report, 2026. URL https://xyz-lab.ai/blogs/ai4ai-at-scale/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. AgentFold: Long-horizon web agents with proactive context management. arXiv preprint arXiv:2510.24699, 2025. URL https://arxiv.org/abs/2510.24699.

Liujie Zhang, Benzhe Ning, Rui Yang, Xiaoyan Yu, Jiaxing Li, Lumeng Wu, Jia Liu, Minghao Li, Weihang Chen, Weiqi Hu, and Lei Zhang. Relax: An asynchronous reinforcement learning engine for omni-modal post-training at scale. arXiv preprint arXiv:2604.11554, 2026. URL https://arxiv.org/abs/2604.11554.

Jiarui Zhao, Rongzhi Zhang, Lingchuan Liu, Hao Yang, Xunliang Cai, and Xi Su. LoHoSearch: Benchmarking long-horizon search agents beyond the human difficulty ceiling. arXiv preprint arXiv:2606.12837, 2026. URL https://arxiv.org/abs/2606.12837.

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. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.22. URL https://aclanthology.org/2025.emnlp-main.22/.

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