AREX: VERIFICATION-GUIDED RECURSIVE REFINEMENT FOR DEEP RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep research requires agents to find answers that jointly satisfy multiple constraints. Discovering such answers is costly, whereas verifying a candidate can often be decomposed into tractable constraint-wise checks. This asymmetry suggests that agents can use the verification results of a provisional answer to direct subsequent research toward the constraints that remain unresolved. We introduce AREX, a verification-guided deep research agent that recursively refines its research state and answer. Its inner research loop gathers evidence, maintains structured records containing source URLs and supporting summaries, and returns a provisional answer, an evidence list, and an answer-level confidence score. The outer refinement loop uses this confidence to accept the result or, when confidence is insufficient, assesses the current trajectory and either launches a targeted refinement round or restarts the investigation. To support long-horizon research, AREX learns an autonomous context-update tool that consolidates its interaction history without relying on an external model. We train dense 4B and 122B-A10B MoE variants on verified synthetic tasks and high-quality trajectories, emphasizing decision-critical steps during both mid-training and reinforcement learning. Under a unified maximum inference budget, AREX achieves strong performance across six research and reasoning benchmarks, while additional recursive rounds yield consistent test-time scaling.

1 INTRODUCTION

Deep research is challenging not only because relevant evidence is difficult to locate, but because a valid answer must often satisfy multiple coupled constraints simultaneously. An agent must discover viable candidates, integrate distributed and potentially conflicting evidence, and verify that each required condition is adequately supported (Yao et al., 2023; Nakano et al., 2021; Huang et al., 2025). Many existing systems address this challenge by extending a single search trajectory with additional reasoning, tool interactions, or context (OpenAI, 2025; Jin et al., 2025; Wu et al., 2025a; Gao et al., 2026). Although greater inference-time computation can broaden exploration, it does not guarantee systematic progress: early errors may persist, exhausted directions may be revisited, and partially valid candidates may be accepted prematurely. The key challenge is therefore not merely to search longer, but to identify which constraints remain unresolved and use that diagnosis to formulate a more targeted next research problem.

We observe that deep research tasks exhibit a fundamental *discovery–verification asymmetry*. Discovering an answer that jointly satisfies all constraints is costly because it requires navigating a large and sparsely informative search space, whereas evaluating a proposed candidate can often be decomposed into substantially simpler constraint-wise checks. Such verification reveals not only whether the current answer is correct, but also which claims are supported, which remain unresolved, and where the available evidence conflicts. Existing work leverages verification either to rank completed candidate trajectories or to refine decisions within an ongoing trajectory (Zeng et al., 2025b; Team et al., 2026). Our key insight is that verification can also define the transition between research rounds. By converting a provisional answer into a partially verified state, the agent can preserve established progress, isolate the remaining uncertainty, and formulate a more targeted next research problem.

We introduce AREX, a deep research agent that recursively refines its research state and provisional answer. AREX alternates between an inner *research loop* and an outer *refinement loop*. The inner

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.

research loop gathers evidence, maintains structured records containing source URLs and supporting summaries, and returns a provisional answer, an evidence list, and an answer-level confidence score. The outer refinement loop uses this confidence to accept the result or continue research. For a low-confidence result, it assesses whether the trajectory contains reusable findings and either formulates a targeted objective for the next round or restarts the investigation. In this way, confidence and trajectory assessment guide the transition between research rounds, rather than serving only as a final filter after search.

Sustaining this recursion over long horizons requires the agent to maintain a concise and actionable research state. During the inner research loop, the interaction history continually accumulates verified evidence, failed queries, speculative hypotheses, duplicated observations, and outdated plans. Retaining the full history can distract subsequent reasoning, whereas indiscriminate truncation may discard evidence needed for later verification (Packer et al., 2023; Hu et al., 2025; Zhou et al., 2026; Wu et al., 2025b; Lu et al., 2026; Zhang et al., 2026). AREX therefore learns to autonomously invoke a dedicated context-update tool during research, converting its own interaction history into a compact research state. This state preserves verified evidence and citations, records constraint-satisfaction status, highlights unresolved information gaps, and specifies the next research plan. Unlike generic summarization or compression performed by an external model, the update is produced by AREX itself and is organized around its current research objective. This autonomous context management keeps the compressed state aligned with the agent's evolving beliefs and future actions.

We train AREX on verified synthetic tasks and high-quality teacher trajectories through multi-stage agentic training and reinforcement learning for long-horizon research. These stages progressively teach the model to search, use tools, construct provisional answers, verify individual constraints, update context, and decide when to continue or terminate. Long research trajectories contain many routine interactions but only a small number of decision-critical steps that substantially affect the final outcome. We therefore identify and emphasize *key steps* across both agentic mid-training and reinforcement learning, including steps where decisive evidence is acquired, incorrect candidates are rejected, or the research direction is revised (Lightman et al., 2024; Wang et al., 2024; 2025; Zeng et al., 2025a).

We instantiate AREX as a dense **4B** model (Turbo) and a Mixture-of-Experts model with **122B total parameters** and **10B activated parameters** (Base). We evaluate them on benchmarks spanning deep search, wide search, multi-constraint information seeking, and reasoning with tools, including BrowseComp (Wei et al., 2025), WideSearch (Wong et al., 2026), DeepSearchQA (Gupta et al., 2026), Humanity's Last Exam (with Tools) (Center for AI Safety et al., 2026), GAIA (Mialon et al., 2024), and xbench-DeepSearch-2510 (xbench Team, 2025). Across these settings, AREX substantially outperforms comparable-scale baselines and remains competitive with models using substantially more activated parameters, demonstrating that recursively improving the research state offers an effective path toward capable and efficient deep research agents.

Our main contributions are summarized as follows:

- We introduce AREX, a recursive deep-research framework that combines structured evidence collection, answer-level confidence and trajectory assessment, and autonomous context updating to accept, refine, or restart the research process.
- We construct verified synthetic research tasks and high-quality trajectories, and develop a multi-stage training pipeline that emphasizes decision-critical steps across agentic mid-training and reinforcement learning.
- We demonstrate strong performance with dense 4B and 122B-A10B MoE models across six benchmarks, together with consistent test-time scaling as the maximum number of recursive research rounds increases.

2 Recursive Research Refinement

2.1 Overall Framework

We adopt Qwen3.5-4B (Qwen Team, 2026) as the backbone model for AREX-Turbo and Qwen3.5-122B-A10B (Qwen Team, 2026) for AREX-Base. As illustrated in Figure 1, AREX implements deep research through a hierarchical, bi-level recursive research refinement process, comprising

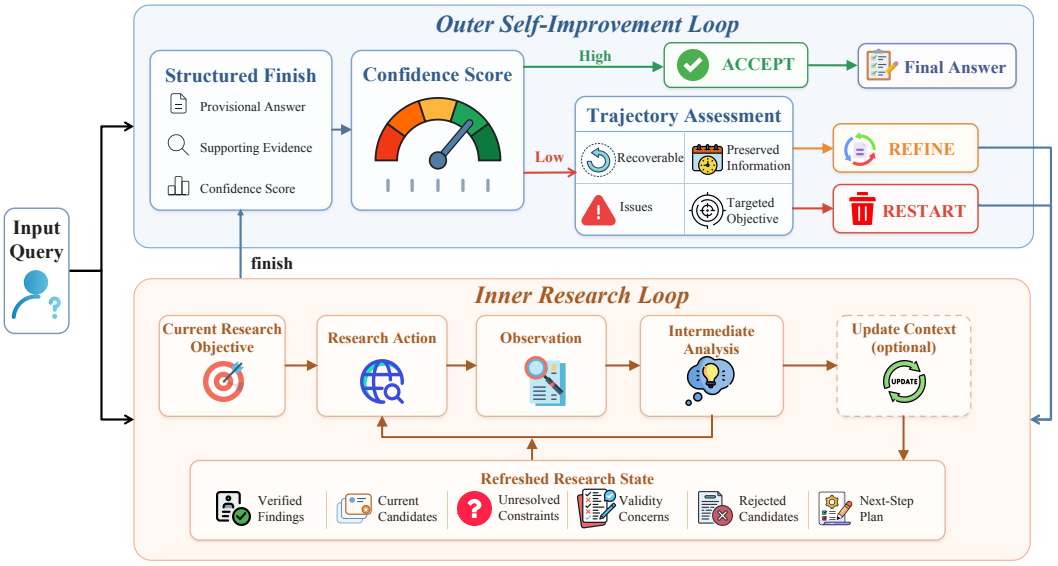


Figure 1: AREX's recursive research refinement framework. The inner research loop maintains the research state and externalizes a provisional answer with supporting evidence and confidence, while the outer refinement loop accepts, refines, or restarts based on confidence and trajectory assessment.

an inner research loop and an outer refinement loop. Given an input query x, it first derives a research objective $q^{(1)}$. The *inner research loop* executes research actions, integrates the retrieved evidence, and updates the current answer until the objective is sufficiently addressed. It then outputs a provisional answer together with supporting evidence and an answer-level confidence score.

The *outer refinement loop* evaluates the provisional result using its confidence score. If the score exceeds a predefined threshold, the answer is accepted. Otherwise, the loop assesses whether the current research trajectory is recoverable: useful findings are preserved and converted into a targeted objective for refinement, while noisy or uninformative trajectories trigger a restart from the original problem.

2.2 Inner Research Loop

The inner research loop incrementally constructs an answer by analyzing the current research objective, invoking search or browsing tools, integrating the returned evidence, and determining the next research step. In the first recursive round, the objective $q^{(1)}$ is derived from the original problem x. In subsequent rounds, the outer loop may provide a targeted objective $q^{(k)}$, such as verifying an unsupported constraint, resolving conflicting evidence, checking temporal validity, or exploring alternatives after a candidate has been invalidated.

Let $h_t^{(k)}$ denote the sequence of intermediate analyses, research actions, and tool observations accumulated up to step t of recursive round k. At each step, the research policy π_θ conditions on the original problem x, the current research objective $q^{(k)}$, and $h_t^{(k)}$ to produce the next analysis and research action. The resulting tool observation is appended to the trajectory, which serves as the model's working research state within the current round.

The research policy adapts as new evidence is accumulated in the trajectory. Supporting observations guide the model toward remaining constraints, whereas contradictory observations may invalidate current candidates and redirect the investigation toward alternatives. When sources conflict, the model seeks evidence with stronger authority, more direct provenance, or greater temporal relevance. If no plausible candidate remains, it may broaden the search space, decompose the objective, or reformulate the search query.

During long-horizon research, the model may invoke `update_context` to consolidate the accumulated trajectory into a more compact research state, allowing subsequent steps to operate on the refreshed context while preserving salient evidence and unresolved constraints. The loop terminates when the current objective has been sufficiently investigated or further search is unlikely to provide substantial benefit. It then invokes `finish` to externalize a provisional answer, its supporting evi-

dence, and an answer-level confidence score. The following two mechanisms describe, respectively, how the trajectory is refreshed during execution and how the terminal trajectory is converted into an answer-level representation for outer-loop evaluation. The complete inner-loop formulation is provided in Appendix D.1.

2.2.1 AUTONOMOUS CONTEXT UPDATING

As research proceeds, the trajectory accumulates search results, intermediate conclusions, rejected candidates, conflicting findings, and evolving plans. Retaining the complete trajectory introduces redundancy and may obscure information relevant to subsequent decisions.

Existing approaches often manage context using fixed heuristics, such as discarding tool responses according to predefined rules (Team et al., 2026) or triggering summarization at a fixed token threshold (Liu et al., 2025). While these strategies reduce context length, they treat context management mainly as a budget-control problem rather than a research-state maintenance problem. As a result, they may remove or dilute decision-relevant information, including source provenance, negative evidence, unresolved constraints, conflicting findings, and the reasons why certain candidates have been rejected.

This limitation becomes more severe in long-horizon research, where the next useful action depends on semantic progress rather than message position or token count. The model must track which constraints have been satisfied, which hypotheses remain viable, and which uncertainties should guide further search. Fixed heuristics are not aligned with these research-state transitions, and may therefore cause the model to revisit invalidated candidates, rediscover prior conclusions, or lose information needed for later refinement.

To address this issue, AREX provides an explicit `update_context` tool. Given the accumulated trajectory $h_t^{(k)}$, the tool constructs a refreshed research state:

$$z_t^{(k)} = f_\theta\left(h_t^{(k)}\right). \tag{1}$$

The refreshed state preserves verified findings and their source identifiers, current candidates, unresolved constraints, validity concerns, rejected candidates, and the next-step plan. Redundant observations, superseded conclusions, and obsolete plans are removed.

After `update_context` is invoked, the refreshed state replaces the preceding trajectory in the model's active context. Subsequent interactions are appended to this state until another update is performed. We denote this combination of the most recent refreshed state and subsequent interactions as the effective context $\bar{h}_t^{(k)}$, on which the research policy conditions when selecting its next action. The complete formulation is provided in Appendix D.2.

The model autonomously determines when to invoke `update_context`, for example after resolving a meaningful subproblem, eliminating a major candidate, reconciling conflicting evidence, or changing its research plan. The tool may therefore be invoked multiple times within an inner loop or not at all.

Rather than generic summarization, autonomous context updating performs trajectory consolidation and research-state refreshing. It enables long-horizon research without requiring the model to repeatedly reconstruct progress from the complete interaction history and allows useful findings and unresolved constraints to be reused across recursive rounds.

2.2.2 STRUCTURED ANSWER EXTERNALIZATION

Autonomous context updating maintains the trajectory during research, but the resulting state is not itself an answer. Once the current objective has been sufficiently investigated, the model invokes a structured `finish` interface.

Let $\bar{h}_{T_k}^{(k)}$ denote the terminal effective context produced by the inner loop, where T_k is the final research step of recursive round k. At recursive round k, the inner-loop output is

$$r^{(k)} = F_\theta\left(\bar{h}_{T_k}^{(k)}\right) = \left(y^{(k)}, \mathcal{E}^{(k)}, s^{(k)}\right), \tag{2}$$

where T_k is the final step, $y^{(k)}$ is the provisional answer to the original problem, and $\mathcal{E}^{(k)} = \{(u_j, e_j)\}_{j=1}^{N_k}$ is an evidence list in which each entry contains a source URL u_j and a summary e_j of the supporting evidence. The scalar $s^{(k)} \in [0, 100]$ is the answer-level confidence score produced by the model.

The confidence score reflects the estimated completeness, consistency, provenance, and temporal validity of the answer. Unlike the full trajectory, which may include discarded directions, obsolete plans, and raw tool responses, $r^{(k)}$ retains only the answer-level information needed for outer-loop evaluation.

Invoking `finish` terminates the current inner loop rather than the entire recursive process. The outer loop evaluates the provisional answer, evidence, and confidence score against the original problem, and either accepts the answer or formulates a targeted objective for another research round.

2.3 Outer Refinement Loop

At recursive round k, the outer self-improvement loop receives the structured result $r^{(k)}$ together with the effective terminal context $\bar{h}_{T_k}^{(k)}$. It then decides whether to terminate the recursive process, refine the current trajectory, or restart the investigation.

The outer loop uses $s^{(k)}$ as the primary signal for deciding whether further research is required. The score reflects the model's overall assessment of the completeness, consistency, provenance, and temporal validity of the provisional answer, while the supporting evidence and detailed research state remain available in $\mathcal{E}^{(k)}$ and $\bar{h}_{T_k}^{(k)}$. Let τ denote the confidence threshold. If $s^{(k)} \geq \tau$, the current answer is accepted and returned as the final answer. Otherwise, the model evaluates whether the current research trajectory contains useful information that can support further improvement.

For a low-confidence result, the trajectory assessment is defined as

$$g^{(k)} = G_\theta \left(x, r^{(k)}, \bar{h}_{T_k}^{(k)} \right) = \left(v^{(k)}, \mathcal{P}^{(k)}, \mathcal{I}^{(k)}, q^{(k+1)} \right), \tag{3}$$

where $v^{(k)} \in \{0, 1\}$ indicates whether the trajectory is recoverable, $\mathcal{P}^{(k)}$ contains information that should be preserved, $\mathcal{I}^{(k)}$ contains issues that require further investigation or revision, and $q^{(k+1)}$ is the research objective for the next recursive round.

The complete decision rule is

$$d^{(k)} = \begin{cases} \textsc{Accept}, & s^{(k)} \geq \tau, \\ \textsc{Refine}, & s^{(k)} < \tau \ \wedge \ v^{(k)} = 1, \\ \textsc{Restart}, & s^{(k)} < \tau \ \wedge \ v^{(k)} = 0. \end{cases} \tag{4}$$

When Refine is selected, the next round is initialized from a refreshed state that preserves the reusable findings in $\mathcal{P}^{(k)}$ while emphasizing the unresolved issues in $\mathcal{I}^{(k)}$. When Restart is selected, the previous trajectory is discarded and the next round is initialized solely from the original problem. The complete state-transition formulation is provided in Appendix D.3.

The procedure runs for at most K outer-loop iterations, each comprising one inner research round followed by one outer-loop decision. Thus, $K = 1$ executes one research round and its corresponding decision but permits no further refinement or restart. For $K > 1$, low-confidence results may trigger additional rounds; if none reaches the confidence threshold, the answer with the highest confidence is returned.

3 Training Data Construction

To train AREX for recursive research, we construct a dataset of challenging research tasks and evidence-grounded interaction trajectories. The pipeline consists of two stages: recursive research task synthesis and teacher trajectory collection with quality control.

3.1 Recursive Research Task Synthesis

We construct three categories of tasks: browse-intensive problems requiring cross-source information synthesis, reasoning-intensive problems requiring multi-step planning or deduction, and scientific-literature problems requiring evidence integration across papers. For each category, human experts define templates specifying the answer format, available sources, reasoning requirements, and verification criteria. Instances are then generated from real-world sources, including webpages, scientific literature, structured knowledge bases, and public repositories.

Given a latent answer y, we identify a set of verifiable constraints $\mathcal{C}(y) = \{c_1, \ldots, c_n\}$, covering properties such as temporal relations, numerical attributes, entity relations, and evidence requirements. We transform these constraints into indirect descriptions to prevent direct keyword matching and require multi-hop research. A task is retained only if the answer is not directly exposed by the query, every constraint can be verified from available evidence, and the joint constraints uniquely identify the answer.

We further check correctness, uniqueness, evidence availability, and difficulty. Tasks with ambiguous answers, inconsistent constraints, or insufficient evidence are removed, while independent research rollouts are used to filter tasks that are either solvable through shallow retrieval or remain unsolved after extensive investigation.

3.2 Teacher Trajectory Collection and Quality Control

For each synthesized task, strong teacher models interact with the same research tools and environment used by AREX. Each collected trajectory records model actions, tool calls, retrieved observations, intermediate analyses, and the final structured answer, thereby exposing the behaviors required for long-horizon research.

We apply quality control at both the interaction and outcome levels. First, we remove trajectories that guess the answer directly, ignore retrieved observations, fail to revise invalid hypotheses, or otherwise lack meaningful iterative investigation. Second, we discard trajectories containing invalid tool executions or unsupported references. Third, we require the final answer to be reconstructable from the collected evidence and remove outputs containing unsupported or contradictory claims. Finally, confidence is used as an auxiliary filtering signal after these validity checks, and trajectories below a predefined threshold are discarded.

The retained trajectories supervise iterative evidence acquisition, research-state maintenance, evidence-grounded answer construction, and adaptive continuation within the recursive research framework.

4 Training Pipeline

4.1 Multi-stage Agentic Mid-training

We adopt a progressive agentic mid-training pipeline with three phases. First, browse-intensive multi-round trajectories establish web navigation, tool use, evidence acquisition, and iterative research behavior. Second, expert reasoning trajectories strengthen long-form reasoning, hypothesis verification, and difficult problem solving. Finally, mixed-capability consolidation combines challenging academic and knowledge-intensive tasks with selective replay from browse-intensive trajectories. This ordering progressively integrates agentic and reasoning capabilities while reducing interference between heterogeneous training objectives.

Key-Step-Focused Consolidation. Long research trajectories contain many routine interactions but relatively few decision-critical steps. Using high-precision rule-based detectors, we identify three types of key steps: acquiring answer-relevant evidence, rejecting an incorrect candidate and redirecting the search, and updating the context after meaningful progress. Annotations are retained only for trajectories that pass final-answer verification and are grounded in valid tool observations.

For each selected step, we preserve its complete preceding context but apply the training loss only to the selected response. This concentrates supervision on consequential research decisions without

discarding the trajectory state required to predict them. Detector definitions, the formal objective, and step-level loss analysis are provided in Appendix E.

4.2 Step-aware Reinforcement Learning

We extend group-relative policy optimization to account for the heterogeneous structure of long research trajectories. Token-level policy ratios are aggregated into length-normalized step-level ratios, and the clipped objective is averaged first across steps and then across trajectories, preventing long responses or trajectories with more interactions from dominating optimization.

All steps receive the group-relative outcome advantage, while detected key steps receive a bounded auxiliary bonus only when the final trajectory is successful. Final-answer correctness therefore remains the primary optimization signal, while the auxiliary term emphasizes decision-critical actions shared with the mid-training stage. We additionally apply a KL penalty against the reference policy. The complete objective and derivation are provided in Appendix F.

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate AREX on six benchmarks spanning four complementary regimes of search-augmented reasoning. **BrowseComp** and **DeepSearchQA** emphasize deep research, requiring multi-step retrieval, query reformulation, evidence aggregation, and answer synthesis. **GAIA** and **xbench-2510** additionally stress agentic task completion, where information seeking must be coordinated with planning, tool use, and multi-step reasoning. **WideSearch** measures broad-coverage retrieval and synthesis over a large search space; we report results on its English subset. Finally, **HLE with tools** evaluates high-level reasoning with access to web search and computational tools. We report the Item-F1 score for **WideSearch**, F1 score for **DeepSearchQA** and accuracy for all other benchmarks.

Evaluation protocols. We evaluate our models with a unified long-horizon search-agent interface comprising `search`, `visit`, `update context`, and `finish` tools. For **HLE with tools**, we add a `python` tool. The agent can iteratively retrieve webpages, inspect sources, maintain a compact working state, and return a structured final response. Following Team et al. (2026), we evaluate both AREX variants with a maximum budget of $300 \times 5 = 1{,}500$ tool calls.

5.2 Overall Performance

As shown in Table 1, AREX achieves strong and consistent performance across diverse search-augmented reasoning tasks. When accounting for model scale, AREX-Base demonstrates a strong capability-to-parameter trade-off with 10B active parameters. It consistently improves over the Qwen3.5 backbone family, including the substantially larger Qwen3.5-397B model, and remains competitive with leading open-source and proprietary research agents across all evaluations. Beyond the Qwen3.5 family, AREX-Base also compares favorably with specialized research agents and other frontier open-source systems. It outperforms MiroThinker-H1 on DeepSearchQA and text-only HLE while remaining within one point on xbench-2510; it also surpasses DeepSeek-V4-Pro on DeepSearchQA, WideSearch-en, and text-only HLE, and exceeds Kimi-K2.6 on GAIA and WideSearch-en. Against proprietary frontier systems, AREX-Base achieves the best reported WideSearch-en score and remains competitive on both BrowseComp and DeepSearchQA. These gains extend beyond search-intensive tasks to broad information synthesis, long-horizon agentic tasks, and expert-level reasoning, with AREX-Base achieving the best overall result on WideSearch-en. A similar trend holds at the compact scale, where the 4B AREX-Turbo outperforms Qwen3.5-35B on five of the six benchmarks. Taken together, these results demonstrate that AREX possesses strong and broadly transferable research capabilities across diverse search, reasoning, and tool-use settings. Its favorable performance at both model scales further demonstrates the effectiveness of our framework, which recursively preserves verified progress and redirects search toward unresolved constraints to enable systematic progress over long research horizons.

Model	BrowseComp	GAIA	xbench-2510	DeepSearchQA	WideSearch-en	HLE (tool)
Frontier models						
GPT-5.4	82.7	–	–	88.5	77.5	52.1*
Opus-4.6	83.7	–	–	91.3	77.5	53.0*
Gemini-3.1-Pro	85.9	80.6	53.0	93.3	66.4	51.4*
Open-source models						
GLM-5	75.9	70.0	–	–	69.8	50.4
Kimi-K2.6	83.2	80.6	90.0	92.5	80.8	54.0*
DeepSeek-V4-Flash	73.2	–	69.0	90.6	76.4	45.1
DeepSeek-V4-Pro	83.4	–	80.0	88.7	78.0	48.2
Tongyi-DeepResearch-30B	43.4	70.9	55.0	–	–	32.9
Qwen3.5-35B	61.0	80.0	50.3	68.5	57.1	47.4
Qwen3.5-122B	63.8	81.6	–	–	60.5	47.5
Qwen3.5-397B	78.6	83.5	61.0	82.1	74.0	48.3
MiroThinker-1.7-mini	67.9	80.3	57.2	67.9	–	36.4
MiroThinker-1.7	74.0	82.7	62.0	72.1	–	42.9
MiroThinker-H1	88.2	88.5	72.0	80.6	–	47.7
Quest-35B	64.6	80.8	–	–	60.6	37.2
Ours						
AREX-Turbo	70.7	81.6	57.0	78.5	68.5	40.6
AREX-Base	82.5	85.4	71.0	89.9	82.0	52.4

Table 1: Comparison results on deep search and agentic reasoning tasks. Models are grouped into frontier closed-source models, open-source models, and our AREX models. * denotes results reported on the full HLE; unmarked results are reported on the text-only subset.

5.3 Inference Framework Analysis

At inference time, AREX executes the inner research loop under a fixed interaction budget. The model uses `python`, `search`, and `visit` to acquire and verify evidence, `update_context` to consolidate the evolving research trajectory into a compact research state, and `finish` to externalize an answer together with its supporting evidence and confidence score. We analyze two mechanisms that determine how AREX uses its inference budget: Autonomous Context Updating (ACU), which refreshes the effective context inside the inner research loop, and the outer refinement loop, which accepts, refines, or restarts a recursive round based on the structured result produced by `finish`. The analysis separates three questions: when the model invokes `update_context`, how ACU and the outer refinement loop change accuracy under controlled comparisons, and whether the answer-level confidence score supports the confidence-based decision procedure. We refer to the complete system as AREX w/ ACU. In the corresponding ablation, denoted as AREX w/o ACU, the same model retains all other tools and operates under the same interaction budget, but cannot refresh its research state and must instead condition on the uncompressed interaction history.

Context-Update Behavior. On BrowseComp, AREX invokes `update_context` in 80.3% of cases, at a mean active-context size of 25,721 tokens, substantially below the 128K-token limit. Most updates are triggered by changes in search strategy or candidate rejection, and the refreshed states most consistently retain unresolved constraints and next-step plans. These observations indicate that context updating is primarily used as a proactive research-state operation rather than as a response to context exhaustion. Complete trigger and content statistics are provided in Appendix G.

Recursive Test-Time Scaling and Autonomous Context Updating. Figure 2(a) evaluates recursive test-time scaling by varying the maximum number of outer-loop iterations K. The $K = 1$ setting executes one inner research round and its corresponding outer-loop decision but permits no additional round, whereas larger values allow low-confidence results to trigger further refinement or restart. Accuracy improves monotonically from 71.41% at $K = 1$ to 82.54% at $K = 5$, with most of the improvement obtained by $K = 3$.

Figure 2(b) evaluates ACU using the same trained model under fixed round budgets. Enabling ACU improves accuracy from 59.6% to 71.4% at $K = 1$ and from 69.8% to 82.5% at $K = 5$. Moreover,

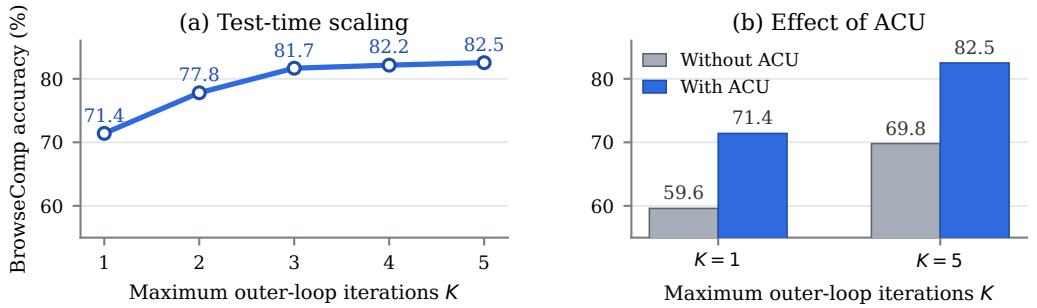


Figure 2: Recursive test-time scaling and autonomous context updating on BrowseComp. **(a)** Accuracy as the maximum number of outer-loop iterations increases from $K = 1$ to $K = 5$. **(b)** Accuracy with and without ACU under the fixed $K = 1$ and $K = 5$ settings. Each inner research round permits at most 300 tool calls.

	Mixed training	**Random-step replay**	**Standard GRPO**	**Full AREX**
Accuracy	77.5	74.1	79.4	82.5
Δ	-5.0	-8.4	-3.1	–

Table 2: Training ablations on BrowseComp. The first three variants respectively replace progressive capability training, key-step focused supervision, and step-aware reinforcement learning.

increasing K improves accuracy both with and without ACU, indicating that autonomous context updating and recursive refinement provide complementary benefits.

5.4 Ablation Studies

We conduct controlled ablations on BrowseComp to evaluate three components of the training recipe. All variants use the same initialization and retain the remaining training pipeline. The corresponding training, supervised-token, optimization, and rollout budgets are matched as applicable.

As shown in Table 2, replacing progressive capability training with mixed training reduces accuracy from 82.5 to 77.5. Replacing key-step focused supervision with equal-budget random-step replay produces the largest drop, from 82.5 to 74.1, indicating that decision-critical steps provide more effective supervision than arbitrary intermediate steps. Finally, replacing step-aware reinforcement learning with standard GRPO lowers accuracy to 79.4. Together, these results show that the three training components provide complementary improvements. Additional step-level loss analysis is provided in Appendix I.1.

6 Conclusion

In this work, we introduced AREX, a deep research agent that recursively refines its research state and provisional answer. AREX exploits the discovery–verification asymmetry in deep research to transform provisional answers into partially verified research states, preserving supported evidence while isolating unresolved claims for subsequent targeted investigation. Coupled with autonomous context updating and step-aware training, AREX achieves strong performance across deep search, wide search, agentic reasoning, and tool-use benchmarks. Our preliminary training analysis further highlights the importance of identifying and reinforcing decision-critical steps in long-horizon trajectories, rather than treating all intermediate actions uniformly. Future work will investigate more general and autonomous mechanisms for estimating step utility and assigning fine-grained training signals. These results suggest that verification-guided state refinement and step-aware optimization offer a promising direction for building reliable long-horizon research agents. The current study focuses primarily on benchmark-level outcome accuracy; broader evaluation of confidence calibration, computational efficiency, and robustness across research environments remains future work.

AI USE STATEMENT

Generative AI models were used as components of the data-construction pipeline to synthesize research tasks and collect teacher trajectories, as described in Section 3. AI tools were also used to assist with language polishing and manuscript presentation. All model-generated training data were subject to the filtering and verification procedures described in the paper. The authors reviewed all AI-assisted content, verified the reported experimental results and scientific claims, and take full responsibility for the content of this submission.

ETHICS STATEMENT

Our training data are constructed from publicly accessible webpages, scientific literature, structured knowledge bases, and public repositories. Although we apply evidence and trajectory filtering, these sources may contain inaccuracies or biases that can be inherited by the resulting models. Moreover, AREX may still produce incomplete or unsupported conclusions and should not be used without appropriate human verification in high-stakes settings.

REPRODUCIBILITY STATEMENT

The main paper describes the data-construction pipeline, training procedure, recursive inference framework, and evaluation protocol. Appendix B specifies the structured output interface, confidence threshold, and inference budget. Appendices D, E, and F provide the complete inference formulation, key-step supervision procedure, and reinforcement-learning objective, respectively. Additional analyses and numerical results are reported in the remaining appendices.

REFERENCES

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. https://doi.org/10.1038/s41586-025-09962-4. Preprint title: Humanity's Last Exam; arXiv:2501.14249.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. doi: 10.48550/arXiv.2412.19437. https://arxiv.org/abs/2412.19437.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. https://jmlr.org/papers/v23/21-0998.html.

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. In *The Fourteenth International Conference on Learning Representations*, 2026. https://openreview.net/forum?id=JSYCLJcniH.

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. doi: 10.48550/arXiv.2601.20975. https://arxiv.org/abs/2601.20975.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32779–32798, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1575. https://aclanthology.org/2025.acl-long.1575/.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Huichi Zhou, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. Deep research

agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025. doi: 10.48550/arXiv.2506.18096. https://arxiv.org/abs/2506.18096.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. doi: 10.48550/arXiv.2401.04088. https://arxiv.org/abs/2401.04088.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *The Second Conference on Language Modeling*, 2025. https://openreview.net/forum?id=Rwhi91ideu.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. https://openreview.net/forum?id=qrwe7XHTmYb.

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*, 2025. doi: 10.48550/arXiv.2507.02592. https://arxiv.org/abs/2507.02592.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. https://openreview.net/forum?id=v8L0pN6EOi.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Beyond the context window: Scaling agentic RL via end-to-end optimized context compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21074–21125, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.966. https://aclanthology.org/2026.acl-long.966/. Earlier preprint: arXiv:2510.06727.

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

MindDR Team and Li Auto Inc. Mind deepresearch technical report. *arXiv preprint arXiv:2604.14518*, 2026. doi: 10.48550/arXiv.2604.14518. https://arxiv.org/abs/2604.14518.

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. doi: 10.48550/arXiv.2112.09332. https://arxiv.org/abs/2112.09332.

OpenAI. Introducing deep research. OpenAI technical post, February 2025. https://openai.com/index/introducing-deep-research/. Official technical webpage; not a scholarly paper.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560. https://arxiv.org/abs/2310.08560.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. 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. In *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551, 2023. https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

MiroMind Team, S Bai, L Bing, L Lei, R Li, X Li, X Lin, E Min, L Su, B Wang, et al. Mirothinker-1.7 & h1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*, 2026.

Tongyi DeepResearch Team et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025. doi: 10.48550/arXiv.2510.24701. https://arxiv.org/abs/2510.24701.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. https://aclanthology.org/2024.acl-long.510/.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Monica Lam, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. doi: 10.48550/arXiv.2504.20073. https://arxiv.org/abs/2504.20073.

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. doi: 10.48550/arXiv.2504.12516. https://arxiv.org/abs/2504.12516.

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

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, pages 120957–120985, 2025a. https://papers.nips.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, Ding-Chu 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. doi: 10.48550/arXiv.2509.13313. https://arxiv.org/abs/2509.13313.

xbench Team. xbench-DeepSearch-2510. Official benchmark and dataset webpage, October 2025. https://xbench.org/agi/aisearch. Versioned benchmark/dataset; no standalone scholarly paper was identified as of 2026-07-12.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, et al. Quest: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=WE_vluYUL-X.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via turn-level credit assignment. *arXiv preprint arXiv:2505.11821*, 2025a. doi: 10.48550/arXiv.2505.11821. https://arxiv.org/abs/2505.11821.

Weihao Zeng, Keqing He, Chuqiao Kuang, Xiaoguang Li, and Junxian He. Pushing test-time scaling limits of deep search with asymmetric verification. *arXiv preprint arXiv:2510.06135*, 2025b.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19149–19164, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.956. https://aclanthology.org/2026.findings-acl.956/.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *The Fourteenth International Conference on Learning Representations*, 2026. https://openreview.net/forum?id=XY8AaxDSLb.

A Related Work

Tool-Augmented Deep Research. Language-model agents extend LLMs from static response generation to iterative problem solving with tools, web environments, and external knowledge sources (Yao et al., 2023; Schick et al., 2023; Nakano et al., 2021). Deep research pushes this paradigm further: agents must locate sparse evidence, compare conflicting sources, and synthesize answers that satisfy multiple coupled constraints (OpenAI, 2025; Huang et al., 2025). Recent systems improve research performance through tool-use supervision, synthetic trajectory construction, agentic fine-tuning, reinforcement learning, and inference-time scaling over longer search traces (Jin et al., 2025; Wu et al., 2025a; Li et al., 2025; Tongyi DeepResearch Team et al., 2025; MindDR Team and Li Auto Inc., 2026; Gao et al., 2026). These approaches primarily strengthen exploration within a research trajectory. AREX addresses a complementary problem: after partial progress has been made, the agent must identify which constraints are already supported, which remain unresolved, and how this diagnosis should define the next research problem.

Verification as Recursive Research Control. Verification has been widely used to improve reasoning, typically as outcome ranking, process supervision, or step-level reward modeling (Lightman et al., 2024; Wang et al., 2024). Recent work further exploits the asymmetry between generating a correct solution and verifying a proposed one, using verification to guide test-time search or critique intermediate agent decisions (Zeng et al., 2025b; Team et al., 2026). AREX uses answer-level assessment as a control signal between research rounds. Answer-level confidence determines whether the current result is accepted or whether further research is required. For low-confidence results, trajectory assessment identifies reusable findings and remaining issues, which are used to refine the next research objective or restart the investigation. This transition-level use complements prior work that applies verification primarily as a final ranking mechanism or a local action critic.

Research-State Management and Long-Horizon Training. Sustained research requires agents to maintain useful information across long trajectories containing evidence, failed searches, speculative hypotheses, and evolving plans. Prior work studies hierarchical memory, virtual context management, learned memory states, periodic summarization, and explicit memory-editing actions (Packer et al., 2023; Hu et al., 2025; Zhou et al., 2026; Wu et al., 2025b; Lu et al., 2026; Zhang et al., 2026; Xie et al., 2026). AREX builds on this direction by learning to invoke a context-update tool that produces an improvement state organized around the current research objective, including verified evidence, citations, constraint status, unresolved gaps, and the next plan. Training such agents also raises sparse credit-assignment challenges, since final success rarely reveals which intermediate steps were decisive. Inspired by process supervision and turn-level credit assignment (Lightman et al., 2024; Wang et al., 2024; 2025; Zeng et al., 2025a), AREX increases exposure to key steps where evidence is acquired, contradictions are resolved, or incorrect directions are repaired, enabling recursive research control at efficient dense and Mixture-of-Experts model scales (Fedus et al., 2022; Lepikhin et al., 2021; Jiang et al., 2024; DeepSeek-AI, 2024).

B Implementation Details

The `finish` tool returns a structured JSON object containing the provisional answer, an evidence list, and an answer-level confidence score. Each evidence entry contains a source URL and a summary of the supporting evidence. The outer refinement loop accepts answers whose confidence reaches the threshold $\tau = 95$; otherwise, it assesses whether the current trajectory should be refined or restarted. For the main evaluation, AREX uses at most $K = 5$ outer-loop iterations, with at most 300 tool calls in each inner research round.

C Additional Benchmark Overview

Figure 3 provides a compact visualization of AREX relative to selected representative baselines across the six evaluated benchmarks. For readability, the figure shows only a subset of the systems included in the main comparison; complete numerical results and evaluation metrics are reported in Table 1.

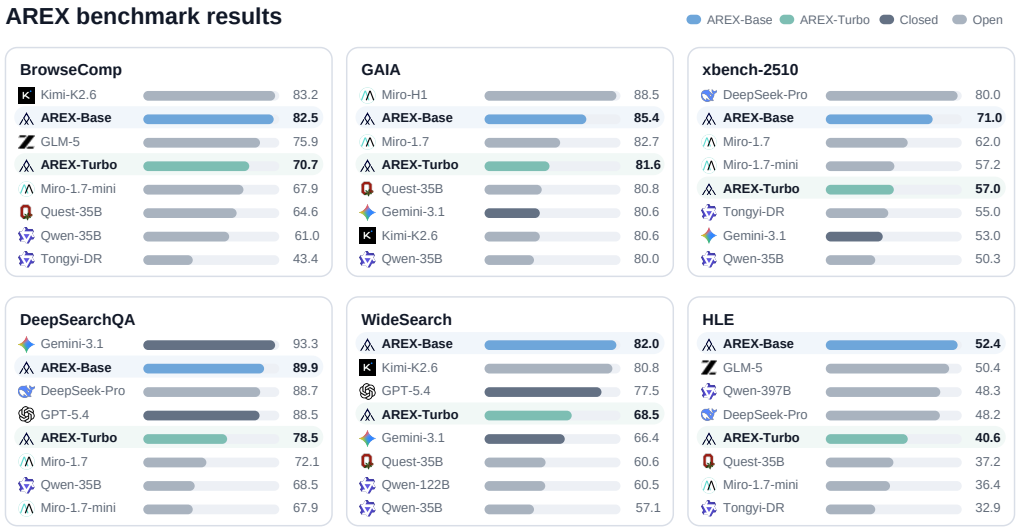


Figure 3: Performance of AREX and selected baselines across six research and reasoning benchmarks. Detailed results and evaluation metrics are reported in Table 1.

D Complete AREX Formulation

D.1 Detailed Inner-Loop Formulation

At recursive round k, the inner research loop operates on the original problem x and the current research objective $q^{(k)}$. Let $m_i^{(k)}$, $a_i^{(k)}$, and $o_i^{(k)}$ denote, respectively, the model's intermediate analysis, research action, and corresponding tool observation at step i. The interaction trajectory accumulated up to step t is defined as

$$h_t^{(k)} = \left[\left(m_i^{(k)}, a_i^{(k)}, o_i^{(k)}\right)\right]_{i=1}^{t}. \tag{5}$$

Before a context update is invoked, the research policy generates the next analysis and action according to

$$\left(m_{t+1}^{(k)}, a_{t+1}^{(k)}\right) = \pi_\theta\left(x, q^{(k)}, h_t^{(k)}\right), \qquad o_{t+1}^{(k)} = \mathcal{T}\left(a_{t+1}^{(k)}\right), \tag{6}$$

where π_θ is the research policy and $\mathcal{T}$ denotes the external research environment, including search and browsing tools.

The resulting interaction is appended chronologically to the trajectory:

$$h_{t+1}^{(k)} = h_t^{(k)} \oplus \left(m_{t+1}^{(k)}, a_{t+1}^{(k)}, o_{t+1}^{(k)}\right), \tag{7}$$

where $\oplus$ denotes chronological concatenation. When autonomous context updating is used, the policy instead conditions on the effective context defined in Appendix D.2.

D.2 Detailed Context-Update Formulation

During a recursive research round, AREX may invoke `update_context` to consolidate the accumulated trajectory into a compact research state. Given the trajectory $h_t^{(k)}$, the updated state is

$$z_t^{(k)} = f_\theta\left(h_t^{(k)}\right), \tag{8}$$

where f_θ denotes the context-update operation performed by the research model.

Let $t_u \le t$ denote the step of the most recent `update_context` invocation. The effective context available to the model at step t is

$$\bar{h}_t^{(k)} = z_{t_u}^{(k)} \oplus \left[\left(m_i^{(k)}, a_i^{(k)}, o_i^{(k)}\right)\right]_{i=t_u+1}^{t}. \tag{9}$$

Thus, interactions preceding the most recent update are represented by the refreshed state $z_{t_u}^{(k)}$, while subsequent interactions remain available in their original form.

If no context update has occurred in the current recursive round, the effective context is the complete trajectory:

$$\bar{h}_t^{(k)} = h_t^{(k)}. \tag{10}$$

The next analysis and research action are generated from the effective context:

$$\left(m_{t+1}^{(k)}, a_{t+1}^{(k)}\right) = \pi_\theta\left(x, q^{(k)}, \bar{h}_t^{(k)}\right). \tag{11}$$

The model may invoke `update_context` multiple times within a research round. After each invocation, only the most recent refreshed state and the interactions that follow it are retained in the active context.

D.3 Outer-Loop State Transitions

For a low-confidence result, the outer loop produces a trajectory assessment

$$g^{(k)} = \left(v^{(k)}, \mathcal{P}^{(k)}, \mathcal{I}^{(k)}, q^{(k+1)}\right), \tag{12}$$

where $v^{(k)}$ indicates whether the current trajectory is recoverable, $\mathcal{P}^{(k)}$ contains reusable findings, $\mathcal{I}^{(k)}$ describes the remaining issues, and $q^{(k+1)}$ is the research objective for the next round.

When the trajectory is recoverable and REFINE is selected, the next recursive round is initialized as

$$h_0^{(k+1)} = \mathrm{Refresh}\left(\bar{h}_{T_k}^{(k)}, \mathcal{P}^{(k)}, \mathcal{I}^{(k)}\right). \tag{13}$$

The refreshed state retains useful evidence and findings from the previous round while directing subsequent research toward the issues in $\mathcal{I}^{(k)}$ through the objective $q^{(k+1)}$.

When the trajectory is not recoverable and RESTART is selected, the accumulated trajectory is discarded:

$$h_0^{(k+1)} = \mathrm{Init}(x). \tag{14}$$

The next inner research loop therefore begins from the original problem without inheriting evidence or intermediate conclusions from the previous trajectory.

E Key-Step Detection and Supervision

Key-step Focused Mixed-Capability Consolidation. A key challenge in long-horizon research training is that supervision is highly uneven across trajectory steps. Most steps are routine transitions, whereas a small number correspond to critical research decisions, such as discovering answer-relevant evidence, redirecting search after rejecting an incorrect hypothesis, or connecting distributed evidence to the target question. These steps often determine whether the trajectory succeeds.

Full-trajectory training applies supervision to all assistant tokens in a successful trajectory, which can dilute the learning signal from high-value decision points. To address this issue, we introduce key-step focused supervision. We first identify candidate key steps using high-precision rule-based detectors that target semantically meaningful research events, such as:

- the first tool invocation whose results or webpage observations provide evidence for an answer-relevant entity or constraint after multiple exploratory steps;
- the first step where the model rejects previously explored incorrect entities or hypotheses and redirects the search toward a more promising direction;
- key context-update steps, where the model invokes the context-update tool to preserve verified evidence in the improvement state, record unresolved conditions, and prepare the next targeted research plan.

These annotations are grounded in verifiable task structure rather than the model's self-reported reasoning. A step is marked as critical only when a valid tool observation introduces answer-relevant evidence, supports a verified transition toward the solution, or produces an actionable improvement state after evidence accumulation. We retain key-step annotations only for trajectories that pass final-answer verification; merely mentioning an answer-related entity, repeating tool calls, or claiming progress is therefore insufficient without externally grounded evidence and a verified successful outcome. All annotations are constructed offline for training tasks with verifiable reference answers and are not used during evaluation or inference.

To verify whether these detected steps are indeed difficult for the model to learn, we perform a step-level loss analysis after full-trajectory training. For each assistant step s_j in a trajectory, we compute the average token loss:

$$\ell(s_j) = -\frac{1}{|s_j|} \sum_{k=1}^{|s_j|} \log \pi_\theta(a_{j,k} \mid c_{j,k}),$$

where $a_{j,k}$ denotes the k-th assistant token in step s_j, and $c_{j,k}$ is its preceding context. We find that the detected key steps consistently have substantially higher loss than ordinary trajectory steps. This suggests that even after full-trajectory supervision, the model still underfits the most informative and decision-critical parts of the trajectory.

Based on this observation, we construct a key-step training set. For each selected key step, we preserve its full preceding context so that the model is trained under the correct trajectory state. However, the loss is applied only to the key step itself, while the prefix tokens, user messages, and tool observations are masked out. The key-step objective is:

$$\mathcal{L}_{\text{key}} = -\mathbb{E}_{s_j \sim \mathcal{K}} \left[\frac{1}{|s_j|} \sum_{k=1}^{|s_j|} \log \pi_\theta(a_{j,k} \mid c_{j,k}) \right],$$

where $\mathcal{K}$ denotes the set of selected key steps.

This design preserves the conditioning context of the selected step while concentrating the optimization signal on high-value research decisions. Empirically, adding key-step focused supervision after full-trajectory training further improves the model's performance on long-horizon deep research tasks, indicating that selective supervision over informative decision points can improve the effectiveness of selective agentic mid-training.

F Step-Aware Reinforcement Learning Details

F.1 Step-aware Reinforcement Learning

Step-aware Group Policy Optimization Our mid-training analysis shows that long-horizon research trajectories are highly heterogeneous: routine tool-use or transition steps are often easy to imitate, while a small number of decisive steps are much harder and more important for final task success. This observation also motivates our reinforcement learning design. In deep research tasks, a trajectory may contain many rounds of search, browsing, hypothesis update, and evidence verification. Since external tools can return noisy or partially relevant observations, not every action in a successful trajectory should receive the same learning signal.

Standard group-relative policy optimization methods compute a sequence-level reward and propagate the resulting advantage to all generated tokens in the trajectory. However, a sequence-level advantage provides relatively coarse credit assignment for long-horizon tool-use trajectories, where different assistant steps may serve substantially different roles. We adopt a turn-level policy optimization formulation and adapt it to long-horizon deep research through hierarchical step-balanced normalization and key-step shaping signals shared with the mid-training stage.

For each prompt x, we sample G trajectories from the old policy. Let M_i denote the number of assistant steps in the i-th trajectory, and let $L_{i,j}$ denote the number of generated tokens in its j-th step. We construct a step-specific shaped advantage $A_{i,j}$ for each assistant step. All steps in the same trajectory share the group-relative outcome advantage, while annotated key steps receive an additional auxiliary bonus.

For token $a_{i,j,k}$, we define the token-level probability ratio as

$$r_{i,j,k}(\theta) = \frac{\pi_\theta(a_{i,j,k} \mid c_{i,j,k})}{\pi_{\theta_{\text{old}}}(a_{i,j,k} \mid c_{i,j,k})},$$

where $c_{i,j,k}$ is the context before token $a_{i,j,k}$. We then aggregate token-level ratios into a length-normalized step-level policy ratio using the geometric mean:

$$\rho_{i,j}(\theta) = \exp\left(\frac{1}{L_{i,j}} \sum_{k=1}^{L_{i,j}} \log r_{i,j,k}(\theta)\right).$$

The length normalization keeps the policy-ratio scale comparable across assistant steps of different lengths. We then apply hierarchical averaging over steps and trajectories to prevent longer trajectories from dominating the training objective.

We adopt hierarchical normalization by first averaging over assistant steps within each trajectory and then over trajectories within the rollout group:

$$\mathcal{L}_{\text{step}} = -\mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{M_i}\sum_{j=1}^{M_i}\min\left(\rho_{i,j}(\theta)A_{i,j}, \text{clip}\left(\rho_{i,j}(\theta), 1-\epsilon, 1+\epsilon\right)A_{i,j}\right)\right].$$

This prevents trajectories with more interaction steps from dominating the objective solely because of their length and better reflects the heterogeneous structure of long-horizon research trajectories.

We add a KL penalty against a reference policy:

$$\mathcal{L} = \mathcal{L}_{\text{step}} + \beta_{\text{KL}}\mathbb{E}\left[D_{\text{KL}}\left(\pi_\theta(\cdot \mid c) \| \pi_{\text{ref}}(\cdot \mid c)\right)\right].$$

Step Reward Shaping In addition to outcome rewards, we introduce step-level reward shaping to emphasize informative intermediate decisions. This design follows the key observation from mid-training: some steps in a long research trajectory are more decisive than others. The key-step annotations described in Section 4.1 are reused as bounded auxiliary shaping signals.

Let R_i denote the trajectory-level outcome reward for τ_i. We first compute the group-relative outcome advantage:

$$A_i^{\text{out}} = \frac{R_i - \mu_R}{\sigma_R + \epsilon},$$

where μ_R and σ_R are the mean and standard deviation of rewards within the sampled group.

For each assistant step $s_{i,j}$, we define a key-step indicator $B_{i,j}$ based on the key-step annotations described in Section 4.1. To avoid rewarding spurious intermediate behavior, the key-step bonus is applied only when the trajectory-level result is valid:

$$\widetilde{B}_{i,j} = \mathbb{I}[R_i > 0] \cdot B_{i,j}.$$

We directly add a bounded auxiliary bonus to annotated key steps in successful trajectories. The final step-level advantage is defined as:

$$A_{i,j} = A_i^{\text{out}} + \lambda_{\text{key}}\widetilde{B}_{i,j},$$

where λ_{key} controls the strength of step-level shaping.

This formulation preserves the final-answer reward as the main optimization signal while adding a small auxiliary preference for decision-critical steps. The shaping term is not intended to solve general credit assignment for all possible intermediate actions. Instead, it provides high-precision supervision for a subset of research steps that are both identifiable and empirically important. We use this lightweight shaping signal as part of the final AREX reinforcement learning recipe, while retaining final-answer correctness as the primary optimization objective.

Update usage		Call trigger		Update content	
Metric	**Value**	**Category**	**Share**	**Category**	**Share**
Cases w/ update	80.3%	Revise search strategy	66.9%	Verified findings	72.1%
Trigger threshold	128,000	Reject candidate	13.6%	Current candidates	39.2%
Min. context tokens	3,788	Verify evidence/answer	5.3%	Unresolved constraints	95.5%
Mean context tokens	25,721	Identify new lead	7.2%	Validity concerns	14.1%
Median context tokens	25,386	Summarize progress	6.4%	Rejected candidates	81.5%
Max. context tokens	128,591	Other	0.6%	Next-step plan	96.4%

Table 3: Context-update behavior on BrowseComp. Context-token statistics are measured when `update_context` is invoked. Trigger shares are single-label, whereas content shares are multi-label.

G Additional Analysis of Autonomous Context Updating

Context-Update Behavior. We first examine ACU as a runtime behavior rather than only as an architectural component. The agent operates within a 128K-token active context window. Within this budget, AREX may invoke `update_context` when the current research trajectory needs consolidation. If the active context reaches the 128K-token limit, the agent must invoke `update_context` before continuing its research.

Table 3 shows that AREX invokes `update_context` in 80.3% of BrowseComp cases. Updates occur at a mean active-context size of 25,721 tokens and a median of 25,386 tokens, well below the configured 128K-token trigger threshold. Because the context limit is checked after the latest tool observation is appended, the measured context length may slightly exceed the 128K trigger threshold. Only 0.01% of updates occur at or above the limit. This timing indicates that ACU is used mainly as a proactive research operation, while the hard limit acts as a safety constraint rather than the dominant trigger.

The call triggers show where the research policy decides that a compact state is needed. Search-strategy revision accounts for 66.9% of calls, followed by candidate rejection at 13.6%. Identifying new leads, summarizing progress, and verifying evidence or a tentative answer account for smaller shares. The model most often refreshes its state when the retrieval direction has become unproductive or when new evidence changes which candidates remain viable.

The update contents highlight the importance of ACU in preserving decision-relevant research state. Overall, ACU retains most of the effective information needed for subsequent reasoning, especially unresolved constraints and next-step plans, which are the most consistently preserved elements. This suggests that ACU prioritizes information about what remains to be checked, what conditions still need to be satisfied, and what actions should be taken next. It also retains verified findings and rejected candidates, ensuring that later decisions are grounded in established evidence while avoiding previously invalidated paths. In this sense, ACU functions as a compact representation of the research process, organizing it around confirmed evidence, remaining constraints, eliminated alternatives, and planned follow-up actions.

Together, the usage, trigger, and content patterns characterize ACU as a trajectory consolidation and research-state refreshing mechanism for long-horizon research. By updating at meaningful turning points and preserving both positive and negative evidence, ACU helps AREX avoid previously rejected paths and spend the remaining interaction budget on unresolved constraints.

H Additional Inference-Time Analysis

H.1 Answer-Level Confidence Analysis

The inner research loop terminates each round by producing a provisional answer, an evidence list, and an answer-level confidence score. The outer refinement loop uses this score to decide whether the

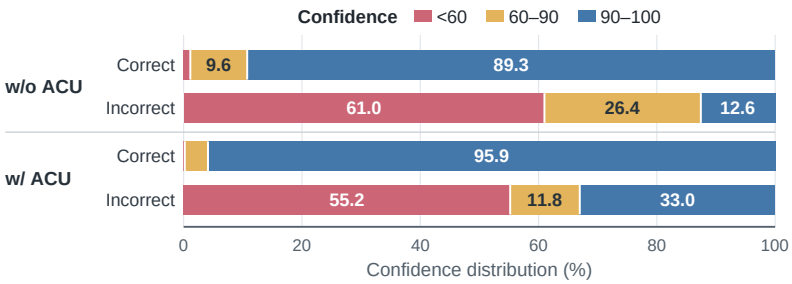


Figure 4: Distributions of answer-level confidence for correct and incorrect BrowseComp outputs, with and without ACU. Each distribution is normalized within its method and outcome group. Confidence intervals are grouped into $[0, 60)$, $[60, 90)$, and $[90, 100]$.

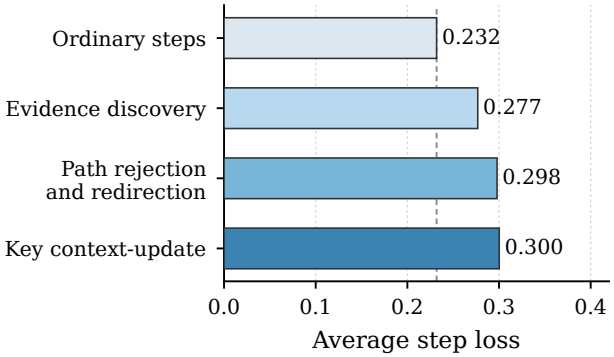


Figure 5: Average step loss after full-trajectory mid-training. Key steps consistently exhibit higher loss than ordinary steps.

current answer should be accepted or whether another research round is required. Figure 4 examines whether the confidence score distinguishes correct from incorrect outputs.

Correct outputs are strongly concentrated in the highest-confidence interval: 89.3% of correct outputs without ACU and 95.9% with ACU receive scores between 90 and 100. In contrast, 61.0% of incorrect outputs without ACU and 55.2% with ACU receive scores below 60. This separation indicates that the answer-level confidence score provides a useful signal for deciding whether to accept the current result or continue research. Although the analysis does not establish probabilistic calibration, it shows that the score discriminates between many successful and unsuccessful outcomes.

I Additional Training Analysis

I.1 Step-Level Loss Analysis

To examine which decisions remain difficult after full-trajectory mid-training, we compute the average token-level negative log-likelihood of each assistant step and compare annotated key steps with ordinary steps.

Ordinary steps have an average loss of 0.232, while evidence discovery, path rejection and redirection, and key context-update steps obtain average losses of 0.277, 0.298, and 0.300, respectively. These values represent relative increases of approximately 19%, 28%, and 29% over ordinary steps. The consistent gap suggests that full-trajectory supervision learns routine actions more readily than decisions that establish evidence, redirect the research process, or consolidate the research state.

This observation motivates concentrating additional supervision on decision-critical steps. Consistent with this interpretation, the controlled ablation in Table 2 shows that replacing key-step focused supervision with equal-budget random-step replay reduces BrowseComp accuracy from 82.5 to 74.1.

Table 4: Exploratory self-distillation experiments on BrowseComp. Both configurations use the 122B-A10B backbone without ACU, the outer refinement loop, or key-step supervision.

Training setting	BrowseComp
Direct training	52.3
Self-distillation	57.1

J Preliminary Exploration under a Simplified Setting

This section investigates a training direction that was not incorporated into the final AREX-Base recipe. The experiment uses an early simplified configuration with a reduced tool set and without ACU, outer refinement loop, key-step supervision, the complete multi-stage data mixture, or the full test-time scaling strategy. Therefore, its absolute BrowseComp scores are not directly comparable with either the complete AREX-Base system or results reported for the original backbone. The comparison is intended only to measure relative differences within the same controlled setting.

Trajectory self-distillation. We conduct a preliminary experiment to examine whether a browse-trained agent can generate improved supervision for a fresh model initialized from the same backbone. Both the direct-training and self-distillation variants use the 122B-A10B initialization, the same problem distribution, and matched training and inference configurations. The direct-training baseline learns from the original browse-intensive trajectories. For self-distillation, an intermediate browse-trained agent regenerates trajectories for the same problems, after which a fresh copy of the base model is trained on the accepted generated trajectories.

Under this simplified setting, trajectory self-distillation improves BrowseComp from 52.3 to 57.1, a gain of 4.8 points (Table 4). This result suggests that trajectories generated by an intermediate agent may provide more effective supervision than the original trajectories for the same backbone. Possible explanations include better alignment with the target policy's action distribution, more consistent search decisions, and cleaner intermediate reasoning structure.

This preliminary study does not isolate the source of the improvement or establish that the gain persists when combined with ACU, key-step supervision, and the complete AREX training recipe. Moreover, self-generated trajectories may inherit or amplify the teacher model's biases and failure modes. We therefore present trajectory self-distillation as a promising direction for future iterations rather than as a component of the final system.