BRING THE ANSWER INTO VIEW: ANSWER-BEARING FRONTIER TRAINING FOR SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome reinforcement learning directly optimizes final-answer correctness, but its terminal signal conflates a trajectory that never acquires the needed information with one that retrieves it but fails to use it. Fine-grained rewards offer denser credit, yet a search trajectory mixes exploration, acquisition, verification, and stopping, for which one proxy need not have a consistent meaning. We study a complementary principle: apply a simple, clear reward at a position where its intended consequence is directly observable. The Answer-Bearing Frontier (ABF) is the state immediately before a tool observation first contains the reference answer; the realized transition witnesses that answer-bearing information is reachable in one step. Answer-Bearing Frontier Training restores these states, samples alternative one-decision continuations, and rewards immediate acquisition, while full-trajectory Outcome updates optimize reaching useful states, integrating information, stopping, and answering. With local supervision only at selected turns, ABF Training achieves 37.4% and 48.8% average exact match across six benchmarks at 2B and 9B, with margins of +2.3 and +1.7 points over the strongest comparators and leads 9 of 12 dataset–scale settings. Relative to Search-R1 at 2B, answer-bearing coverage rises from 53.8% to 61.8%, while exact match given such information changes from 51.9% to 53.0%. The gains extend beyond the answer types used for local training. These results show that a strong training position can turn a narrow local reward into stable end-to-end gains.[1]

1 INTRODUCTION

Reinforcement learning can train search agents by rewarding correct final answers (Jin et al., 2025). This aligns training with task success, but search involves a sequence of retrieval and reasoning decisions. A terminal reward does not directly distinguish how individual decisions contributed to the final outcome.

Process supervision supplements terminal rewards with feedback on intermediate decisions. Existing methods score retrieval relevance, entity coverage, answer likelihood, future success, or search necessity (Hsu et al., 2025; Setlur et al., 2025; Wang et al., 2026; Zhao et al., 2026a; Wu et al., 2026b). A multi-turn trajectory, however, mixes exploration, bridge retrieval, acquisition, verification, and stopping. One local criterion need not carry the same meaning for all these decisions. For example, local rewards can measure increases in the model's likelihood of the reference answer (Wang et al., 2026). Retrieving direct evidence may produce a clear increase. Yet a bridge search may identify the entity needed for the next query without immediately increasing that likelihood.

Applying the same local signal to steps with different objectives can therefore introduce noise into supervision. Work on supervised fine-tuning suggests that carefully selected, high-quality examples can be more effective than larger, less selective datasets (Zhou et al., 2023; Chen et al., 2024b). This motivates a "less is more" perspective on local supervision for search agents. We therefore ask whether a simple local reward can yield greater end-to-end gains when concentrated at positions where it provides clear guidance.

[1]Code, data, and other reproducibility materials: `https://anonymous.4open.science/r/ABF-4B7E/` (anonymous repository).

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

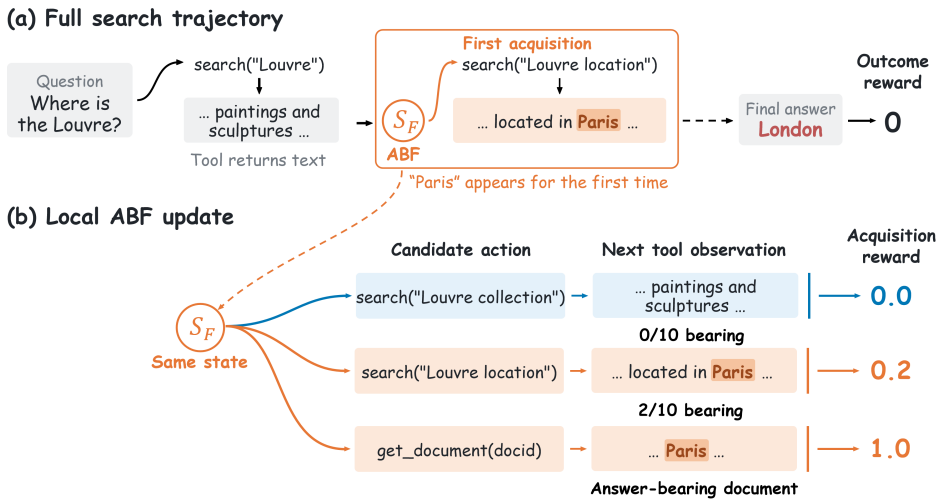


Figure 1: Overview of ABF Training. We restore the state before the first answer-bearing observation and sample new one-decision continuations. ABF updates alternate with full-trajectory Outcome updates; the reference answer is evaluator-only. Local scores omit the format bonus.

Information acquisition is central to effective search. We identify a position where a simple local reward can provide clear supervision for this capability. The reference answer is absent from every observation in a prefix and then first appears in the next tool observation. We call the state immediately before this event the *Answer-Bearing Frontier* (ABF). At this frontier, acquisition is still unresolved, while the observed transition shows that answer-bearing information is reachable in one step. A simple reward can therefore evaluate whether the next action achieves this immediate acquisition objective.

We introduce Answer-Bearing Frontier Training (ABF Training) to strengthen information acquisition at these states. As illustrated in Figure 1, we mine ABFs from complete current-policy trajectories and restore their interaction histories. We then sample new one-decision continuations and reward immediate acquisition of answer-bearing text. These local updates alternate with full-trajectory Outcome updates that reward correct final answers. This combination links targeted acquisition training to end-to-end search performance.

Across six benchmarks, ABF Training improves average exact match over the strongest evaluated baselines by +2.3 points at 2B and +1.7 points at 9B. On the 2B validation set, replacing the frontier with a random answer-bearing position reduces EM by 13.1 points with the reward and schedule fixed (Figure 2(a)), showing the importance of where local supervision is applied. A sampling-cost audit additionally quantifies frontier-mining overhead.

Our contributions are threefold:

- **Method.** We introduce frontier-local acquisition training alternating with full-trajectory Outcome optimization.
- **State selection.** A controlled position ablation demonstrates the value of first-occurrence selection under a fixed local reward.
- **Empirical evidence.** We demonstrate higher average EM across six benchmarks at two scales, increased answer-bearing coverage, and gains beyond training answer types.

2 Related Work

Search agents and complete-trajectory learning. Tool-using language models interleave reasoning and external actions (Yao et al., 2023), and retrieval-augmented models learn retrieval control (Asai et al., 2024). Search-R1 established outcome-only RL over complete search trajectories (Jin et al., 2025). Later work studies training-question selection, local versus global retrieval utility, reward density, and when to search or stop (Jiang et al., 2025; Liu et al., 2026a; Luo et al., 2026; Sun et al., 2026; Java et al., 2026).

Process supervision and credit assignment. Process supervision evaluates intermediate decisions, while reward shaping and return decomposition address delayed credit (Uesato et al., 2022; Lightman et al., 2024; Ng et al., 1999; Arjona-Medina et al., 2019). LeReT uses gold document relevance, while IGPO uses changes in teacher-forced reference-answer likelihood (Hsu et al., 2025; Wang et al., 2026). Rewarding Progress measures changes in future success under a complementary prover (Setlur et al., 2025). DeSA, E-GRPO, HiPRAG, and iStar derive feedback from trajectory-wide answer recall, synthetic entities, external LLM judgments, and learned outcome-ranked preferences, respectively (Wang et al., 2025b; Zhao et al., 2026a; Wu et al., 2026b; Liu et al., 2026b). Other search methods use progress, graph, critic, or self-distilled signals over intermediate turns (Li et al., 2025b; Zhang et al., 2025; Wang et al., 2025a; Li et al., 2025a; Tan et al., 2026; Wu et al., 2026a).

State selection and new branches. TSPO uses the first retrieved occurrence of the reference answer to assign partial credit to prefix turns when the final answer is incorrect (Ma et al., 2026). PivotRL selects states offline from expert trajectories according to local reward variance and difficulty, then trains short continuations using a functional verifier (Yi et al., 2026). Other methods use state or process scores to choose branches for outcome training (Djuhera et al., 2026; Li et al., 2026; Zhang et al., 2026). BranPO truncates current-policy trajectories at shared prefixes and resamples complete suffixes to obtain contrasting final outcomes (Zhao et al., 2026b). ABF Training selects answer-bearing frontiers from current-policy trajectories, rewards immediate acquisition in new one-decision continuations, and alternates these updates with full-trajectory Outcome optimization.

3 Answer-Bearing Frontier Training

In multi-turn search QA, ABF augments Search-R1's final-answer supervision with local rewards for first acquiring answer-bearing information. We define the search setting and Outcome objective, then describe frontier selection, one-decision local updates, and their alternating schedule.

3.1 Search Trajectories and Outcome Optimization

Given a short-answer question q and its reference answer y, the policy π_θ interacts with an external search environment through two tools. `search(query)` returns a ranked list of candidate documents with snippets and document identifiers. `get_document(docid)` opens one candidate and returns its document content. At tool interaction t, the state s_t contains the question and the complete interaction history visible to the model. The policy samples an action $a_t \sim \pi_\theta(\cdot \mid s_t)$, and the environment returns an observation o_{t+1}. After completing its tool interactions, the agent produces a terminal response $\hat{y}$. A complete trajectory is

$$\tau = (s_0, a_0, o_1, \ldots, s_{T-1}, a_{T-1}, o_T, \hat{y}). \tag{1}$$

Our Outcome update follows the outcome-only formulation of Search-R1 (Jin et al., 2025). It starts from the original question, samples a complete trajectory, and uses normalized exact match as its binary task reward,

$$R_{\text{out}}(\tau) = \text{EM}(\hat{y}, y) \in \{0, 1\}. \tag{2}$$

Both Outcome and ABF updates use Group Relative Policy Optimization (GRPO) (Shao et al., 2024); they differ in rollout boundary and reward.

3.2 Identifying the Answer-Bearing Frontier

To apply local supervision selectively, we need a state at which the acquisition objective is both unmet and known to be reachable in one transition. We locate such states in trajectories sampled

by the current policy. We evaluate tool observations using a lexical matching rule $b(o, y)$. It checks the model-visible text of observation o for the canonical answer or a conservatively filtered alias. Matching uses normalized text and token boundaries. We set $b(o, y) = 1$ when a match is found and 0 otherwise, and call matched observations *answer-bearing*. This predicate gives a deterministic acquisition signal over model-visible retrieval results, while final answer quality remains governed by Outcome reward. The rule is fast to evaluate, avoiding the cost and latency of LLM-based scoring or additional sampling for reward estimation. However, lexical matches can be misleading, particularly for numerical and Boolean answers. We therefore restrict training to questions with named-entity answers, such as names of people or places. Lexical occurrences of answers or annotated entities have also been used as rule-based search signals (Wang et al., 2025b; Ma et al., 2026; Zhao et al., 2026a). Appendix C details the matching controls and remaining error modes of this operational proxy.

For a trajectory with at least one answer-bearing observation, we define

$$t_F = \min\{t : b(o_{t+1}, y) = 1\}, \qquad s_F = s_{t_F}, \qquad a_F^{\text{wit}} = a_{t_F}. \tag{3}$$

We call s_F the *answer-bearing frontier* and a_F^{wit} its witness action. No earlier observation in the prefix is answer-bearing, while the recorded witness establishes that at least one action from the same visible state can acquire answer-bearing text in the next transition. Immediate acquisition is thus a concrete, observable local objective at s_F.

After answer-bearing text is already available, rewarding further matches may favor repeated retrieval even when the next useful decision concerns verification or answer integration. Trajectories that never expose answer-bearing text provide no ABF state and remain in the full-trajectory Outcome stream.

3.3 Strict One-Decision ABF Updates

An ABF update trains the next decision at s_F to strengthen information acquisition. For each sampled frontier, we restore its complete model-visible history and sample G new continuations from $\pi_\theta(\cdot \mid s_F)$. Each continuation contains exactly one new policy decision. A tool call is executed once, and the continuation stops after the returned observation. A terminal decision produces a final answer without calling a tool.

The local acquisition reward measures whether the next tool action exposes answer-bearing text. A bridge query that does not yet expose the answer receives no acquisition reward, while its contribution to complete search is optimized by Outcome updates. Direct supervision from retrieval outputs has also been used in query-policy training (Hsu et al., 2025), although that work uses gold document relevance. We define separate rewards for `search`, `get_document`, and terminal responses.

For a `search` continuation, o is the ranked candidate set returned by the environment. Let $u(o; y)$ count its distinct documents with answer-bearing visible text. When $u(o; y) > 0$, let $r(o; y)$ be the rank of the first such document. With retrieval cutoff $K = 10$, the search reward is

$$R_{\text{search}}(o; y) = \frac{u(o; y)}{K} + \alpha \frac{\mathbf{1}[u(o; y) > 0]}{r(o; y)}, \qquad \alpha = 0.01. \tag{4}$$

The reciprocal term is zero when $u(o; y) = 0$. The document-count term rewards broader answer-bearing coverage, while the rank term breaks ties in favor of earlier matches. We place little weight on rank because agent browsing can depend weakly on result position (Zhou et al., 2026).

For a `get_document` continuation, o is the document text returned by the environment. We score whether this document is answer-bearing:

$$R_{\text{doc}}(o; y) = b(o, y). \tag{5}$$

For a terminal response, we use normalized exact match:

$$R_{\text{term}}(\hat{y}; y) = \begin{cases} 2, & \text{EM}(\hat{y}, y) = 1, \\ -2, & \text{EM}(\hat{y}, y) = 0. \end{cases} \tag{6}$$

This reward discourages premature answering and encourages further information acquisition, while still rewarding correct answers.

Training also adds a small format bonus to valid outputs and penalizes invalid outputs. Appendix C.3 gives the complete specification.

GRPO normalizes rewards among continuations from the same restored state, and policy loss is applied only to the newly generated action tokens; the saved prefix and environment result are masked.

3.4 Alternating ABF and Outcome Updates

ABF updates strengthen immediate information acquisition, while Outcome updates optimize complete search trajectories for final-answer correctness. Training proceeds in rounds. Each round consists of N_{ABF} consecutive strict one-decision ABF optimizer steps followed by N_{out} consecutive full-trajectory Outcome optimizer steps from the original questions.

The current policy periodically regenerates trajectories and refreshes the frontier pool. Because naturally mined frontiers concentrate near the start of a trajectory, we stratify the pool by *frontier depth*, the number of tool interactions in the prefix of s_F, and sample fixed quotas across depths.

We monitor answer-bearing coverage on held-out validation questions. Once acquisition gains plateau, we devote the remaining training budget to Outcome updates to focus on final-answer correctness. Appendix A specifies the block lengths, pool refresh, depth quotas, and stopping rule.

4 Experiments

4.1 Experimental Setup

Datasets and metric. We train on the training splits of HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We restrict training to questions with named-entity answers, while validation and evaluation are not restricted by answer type. We evaluate in-domain on HotpotQA, 2WikiMultiHopQA, and MuSiQue, and out-of-domain on InfoSeek (Xia et al., 2025), Bamboogle (Press et al., 2023), and PopQA (Mallen et al., 2023). We report normalized exact match (EM) as the main metric and use the equal-weight mean of the six dataset scores as the overall summary.

Unless otherwise specified, ablations and analyses use Qwen3.5-2B and a held-out validation set of 210 questions, sampled at random in equal numbers from the three training sources.

Baselines. We use **Search-R1** as the outcome-only reference; it optimizes complete search trajectories using terminal exact-match reward (Jin et al., 2025). The other baselines add intermediate supervision at different granularities. **DeSA** applies an answer-recall reward to complete trajectories before subsequent Outcome training (Wang et al., 2025b). **TSPO** assigns turn-level credit from the first retrieved occurrence of the gold answer (Ma et al., 2026), while **IGPO** uses the turn-wise change in the gold answer's log probability (Wang et al., 2026).

Implementation details. We fine-tune the post-trained Qwen3.5-2B and Qwen3.5-9B checkpoints (Qwen Team, 2026). For each model scale and training seed, we create **Base (Outcome-20)** with 20 Outcome warm-up updates and share it across methods within that seed. This warm-up prevents answer-format differences from dominating the small-model comparison. We adapt the baselines' open-source implementations to a shared GRPO training setup (Shao et al., 2024), retaining their respective supervision signals.

All methods share the same prompt, tool interface, and retrieval configuration. Retrieval uses BGE-M3 (Chen et al., 2024a) over the Wiki-25-512 corpus (Luo, 2025). At evaluation, we use greedy decoding, one trajectory per question, and a limit of ten tool interactions. The OOD benchmarks therefore test transfer across question distributions while holding the corpus, retriever, and tool interface fixed.

Table 1 reports ABF and Search-R1 means over three runs with different random seeds at each model scale. DeSA, TSPO, and IGPO each use one run per scale due to cost, but still provide informative comparisons across different supervision approaches. Every run has a separate budget of at most 300 post-Base optimizer steps and independently selects its highest-validation-EM checkpoint under the same rules. Further details are in Appendices A–C and E.

Table 1: Results on in-domain and out-of-domain question distributions (normalized EM, %). ABF and Search-R1 report means over three training runs; other baselines use one run. Avg. weights all six datasets equally. Best and second-best results are shown in bold and underlined.

	In-domain			Out-of-domain			
Method	**HotpotQA**	**2Wiki**	**MuSiQue**	**InfoSeek**	**Bamboogle**	**PopQA**	**Avg.**
			Qwen3.5-2B				
Search-R1	33.1	46.6	9.8	33.3	<u>28.0</u>	50.8	33.6
DeSA	**35.2**	<u>50.4</u>	10.0	<u>34.3</u>	**28.8**	52.0	<u>35.1</u>
TSPO	32.9	46.2	<u>11.0</u>	34.0	27.2	<u>52.2</u>	33.9
IGPO	33.7	40.2	9.6	32.0	24.0	50.6	31.7
ABF Training (ours)	<u>34.7</u>	**53.4**	**12.6**	**43.3**	24.8	**55.4**	**37.4**
			Qwen3.5-9B				
Search-R1	44.8	<u>65.8</u>	18.2	51.0	52.0	49.6	46.9
DeSA	44.5	61.6	18.8	<u>51.3</u>	48.8	**50.4**	45.9
TSPO	<u>45.8</u>	65.2	18.8	50.0	<u>52.8</u>	<u>49.8</u>	<u>47.1</u>
IGPO	41.7	62.6	<u>19.2</u>	50.3	49.6	47.8	45.2
ABF Training (ours)	**46.3**	**68.0**	**21.2**	**53.0**	**54.4**	<u>49.8</u>	**48.8**

4.2 Main Results

ABF improves final-answer accuracy at both model scales. Table 1 shows that ABF Training achieves the highest average EM across the six benchmarks: 37.4% at 2B and 48.8% at 9B. It exceeds the strongest baselines, DeSA at 2B and TSPO at 9B, by +2.3 and +1.7 percentage points, respectively, and leads in 9 of the 12 dataset–scale settings. The gains at both scales indicate that local acquisition training complements Outcome optimization even as the underlying model becomes stronger.

Intermediate supervision does not uniformly improve over Outcome training. Under our shared training setup, DeSA improves average EM over Search-R1 by 1.5 points at 2B but falls 1.0 point below it at 9B. TSPO yields small average gains at both scales, while IGPO scores below Search-R1 at both. ABF produces larger gains of 3.8 and 1.9 points, respectively. This pattern suggests that the benefit depends on how intermediate supervision is designed and applied. The position ablation below further examines frontier selection while keeping the local reward and training schedule fixed.

The gains extend to unseen question distributions. Across InfoSeek, Bamboogle, and PopQA, ABF Training also achieves the highest average EM, exceeding the strongest competing OOD average by +2.8 points at 2B and +1.5 points at 9B. A particularly large gain appears on InfoSeek at 2B, where EM rises from 33.3% with Search-R1 to 43.3% with ABF. Transfer gains are not uniform, however: ABF scores below Search-R1 on Bamboogle at 2B.

4.3 Ablation Studies

We examine two design choices that determine where ABF supervision is applied: first-occurrence selection and depth-aware frontier pool construction. The complementary roles of ABF and Outcome updates are studied through their training dynamics in Section 5.4.

First-occurrence selection is critical to the gains from local supervision. We replace the first answer-bearing transition with an answer-bearing transition sampled uniformly from the same eligible trajectory, keeping state restoration, depth quotas, local reward, group size, and the ABF–Outcome schedule unchanged. The resulting curve largely tracks Search-R1 and finishes 13.1 EM points below the full method in Figure 2(a), losing most of the additional gain from local supervision. Before the answer first appears, the reward targets an unmet acquisition objective. Afterward, useful actions may involve verifying a relation, completing a reasoning chain, or integrating information into an answer. Rewarding more answer-bearing documents at these later states may encourage redundant retrieval without distinguishing such progress. The ablation thus supports selecting a boundary where the local reward has a clear acquisition purpose.

Depth-aware pool construction preserves deeper acquisition opportunities. Without stratification at 2B, 85.0% of selected states fall in the first two frontier positions. Using displayed scores,

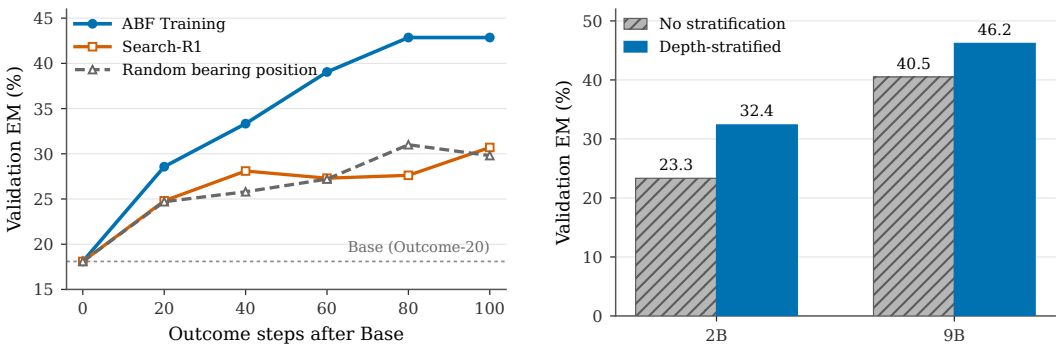


Figure 2: **Ablations of ABF state construction.** (a) compares the first answer-bearing transition with a uniformly sampled answer-bearing transition from the same eligible trajectory. (b) compares direct and depth-aware pool construction at 2B and 9B.

stratification improves EM by 9.1 points at 2B and 5.7 at 9B in Figure 2(b). Across the evaluation benchmarks, removing stratification has little effect on InfoSeek, whose questions combine multiple constraints to identify an entity (Xia et al., 2025). The drop is larger on MuSiQue, which is built from connected 2–4-hop questions (Trivedi et al., 2022). The larger effect on MuSiQue is consistent with the value of preserving training opportunities for further acquisition after intermediate information has been gathered.

5 Analysis

5.1 ABF Improves Evidence Acquisition

ABF improves both initial search and overall evidence acquisition. Table 2 compares acquisition behavior across evaluation sets. ABF's initial queries retain more question terms, and the first-search answer-bearing hit rate increases from 25.1% to 31.3% over Search-R1. These queries are formed before any retrieved observation is available, indicating better initial query formulation.

Table 2: **Statistics of evaluation trajectories.**

	First search transition		Complete search trajectory			
Policy	**Question-term retention (%)**	**Bearing any-hit (%)**	**Search any-hit (%)**	**Bearing mass / rollout**	**Bearing open rate (%)**	**Completion tokens / turn**
Search-R1	53.4	25.1	21.0	1.0	24.8	34.7
ABF Training	**73.5** (+20.1)	**31.3** (+6.2)	**25.8** (+4.8)	**1.7** (+0.7)	**32.1** (+7.3)	**51.5** (+16.8)

Note. Statistics are averaged equally across datasets; bearing metrics exclude Boolean questions. Question-term retention is the fraction of question content terms retained in the first query. Bearing any-hit is the fraction of first searches returning at least one answer-bearing document. Search any-hit is the corresponding fraction over all searches. Bearing mass is the average number of answer-bearing documents per rollout. Bearing open rate is the fraction of document opens that expose answer-bearing text. Completion tokens per turn is the average number of tokens generated in an assistant turn.

The gains also hold across complete trajectories. The fraction of searches returning answer-bearing results rises from 21.0% to 25.8%, while the fraction of document opens exposing answer-bearing text rises from 24.8% to 32.1%. Thus, the greater amount of answer-bearing information per rollout is accompanied by higher hit rates for individual tool calls, spanning both search and document access.

Inspection of the search trajectories shows that ABF tends to preserve more question constraints in its initial queries and use intermediate findings to refine subsequent searches. Detailed case comparisons are provided in Appendix D.

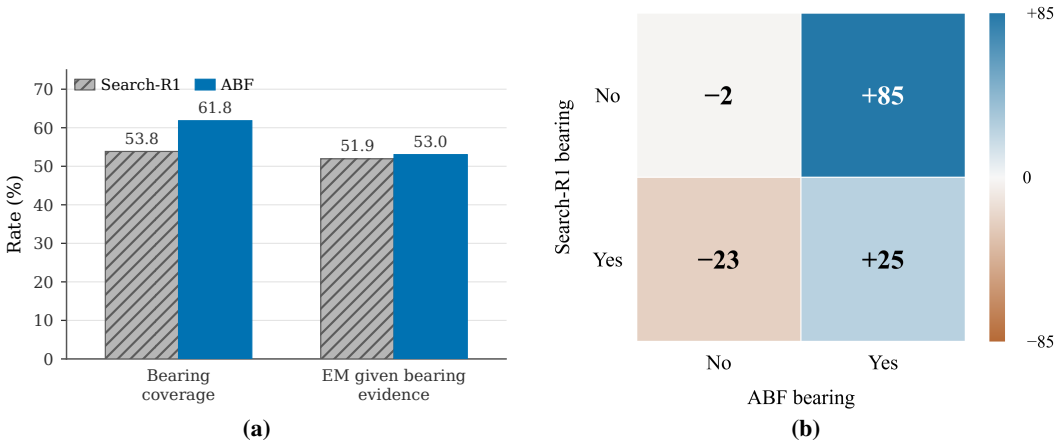


Figure 3: **From acquisition to final answers.** On 2,032 non-Boolean evaluation questions from HotpotQA, 2WikiMultiHopQA, MuSiQue, and InfoSeek, (a) shows equal-dataset mean answer-bearing coverage and EM given bearing text; (b) shows paired net correct-answer changes by bearing transition, pooled across datasets. The highlighted No→Yes cell contributes a mean net gain of +85; the other cells sum to zero at the displayed precision.

5.2 Acquisition Changes and Final Answers

Final-answer gains align with improved evidence acquisition. We next examine how acquisition changes relate to final-answer accuracy on the same evaluation questions. In Figure 3(a), ABF increases bearing coverage from 53.8% to 61.8%, while EM among questions with bearing text changes from 51.9% to 53.0%. This pattern suggests that broader access to answer-bearing information contributes to the overall gain.

Figure 3(b) examines this relationship through paired question-level changes. The No→Yes bearing group contributes a mean net gain of 85 correct answers, matching the mean total non-Boolean gain; mean changes in the other three groups approximately cancel. This is consistent with acquisition improvements contributing to final-answer gains.

5.3 Generalization Beyond Entity Answers

Despite training only on named-entity answers, ABF improves all reported non-entity categories (Table 3): 3.4 token-F1 points overall versus 2.4 for entities, with the largest gains on Yes/No (+5.1) and other text/list answers (+3.9). Together with the acquisition analyses, this supports transfer beyond training answer types.

Table 3: **Generalization across answer types (token F1).** Results on evaluation questions with answer-type annotations. Token F1 gives partial credit for token overlap in multi-token answers.

Answer type	Search-R1	ABF (Δ)
Named entity	49.3	51.7 (+2.4)
Non-entity (all)	**36.7**	**40.1** (+3.4)
Yes/No	79.6	84.7 (+5.1)
Number/quantity	27.8	29.7 (+1.9)
Date/time	35.5	38.4 (+2.9)
Other text/list	25.4	29.3 (+3.9)

5.4 ABF Expands Search; Outcome Consolidates It

ABF and Outcome play complementary roles. ABF encourages acquisition and discourages premature answering, which can increase tool use. At the ten-tool limit, agents must answer; an incorrect forced answer receives no correctness reward. Outcome thus indirectly discourages prolonged, unsuccessful search. The oscillating rollout lengths are consistent with a balance between expanding retrieval opportunities and consolidating a more selective search policy through final-answer feedback.

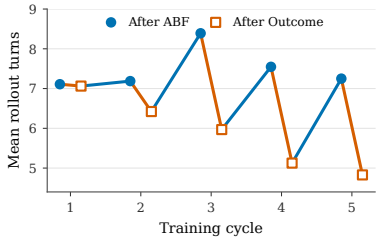


Figure 4: **Alternating ABF and Outcome updates.** Mean full training-rollout length after each block. Training alternates ABF and Outcome blocks.

Figure 4 shows a clear expansion–compression pattern from the second cycle onward: ABF re-expands rollouts and Outcome compresses them again. Post-Outcome length falls from 7.06 turns in cycle one to 4.83 in cycle five, consistent with improved acquisition enabling more effective evidence use and earlier completion.

5.5 Estimated Sampling Cost

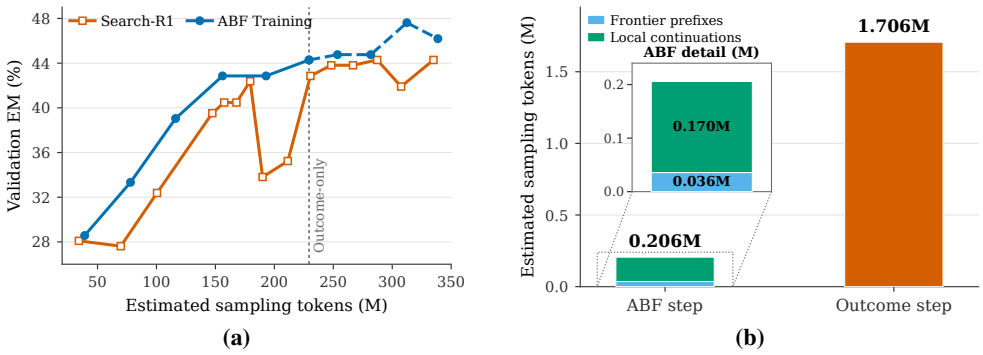


Figure 5: **Selected-prefix sampling estimates.** (a) Validation EM; dashed: Outcome-only tail. (b) Tokens per step. ABF excludes unselected mining and source suffixes; see Appendix E.

Rollout generation is a major RL training bottleneck (Kim et al., 2026); we report cumulative sampling tokens (Zong et al., 2026). These count model-call inputs and outputs for complete Outcome rollouts and selected ABF prefixes through the first answer-bearing observation plus eight new one-decision continuations per frontier. Unselected mining, source suffixes, optimizer updates, initialization, and validation are excluded.

Under this selected-prefix accounting, ABF reaches 47.6% validation EM at 312M tokens versus Search-R1's 41.9% at 308M (Figure 5(a)); its displayed Outcome-only tail remains higher over the shared token range. Per-step sampling is 12.1% of paired Outcome (0.206M/1.706M; Figure 5(b)). Adding threefold-weighted update-sequence tokens gives 0.503M/2.608M, or 19.3% under this selected-prefix work proxy. Appendix E reports both metrics including full mining.

6 Conclusion

ABF Training alternates frontier-local acquisition updates with full-trajectory Outcome optimization, achieving the highest six-benchmark average EM among evaluated methods at 2B and 9B. Greater bearing coverage and gains beyond training answer types are consistent with improved acquisition. Random bearing-position selection largely removes the local-supervision gains under the same reward and schedule. These findings highlight where to supervise as a design choice alongside what to reward.

AI USE STATEMENT

The authors led the research and made the final methodological and reporting decisions. Generative AI tools assisted code implementation, provided feedback on methodology and experimental reporting, and helped interpret results, including sampling-cost accounting. They also assisted language editing, literature lookup, manuscript organization, and formatting and consistency checks. The authors reviewed the AI-assisted work and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendices A–E record the training procedure, data roles, agent interface, reward and optimization settings, evaluation protocol, sampling-cost accounting, and qualitative trajectories. We have publicly released our code, data, and supporting materials, including the sources used to generate the tables and figures, through the anonymous repository linked on the first page.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/16105fb9cc614fc29e1bda00dab60d41-Abstract.html.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024a. URL https://arxiv.org/abs/2402.03216. Version v5, 12 December 2025.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. AlpaGasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9543942c237ded1b39b1fd37259ff88e-Abstract-Conference.html.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, Heiko Ludwig, and Holger Boche. TSR: Trajectory-search rollouts for multi-turn RL of LLM agents. *arXiv preprint arXiv:2602.11767*, 2026. URL https://arxiv.org/abs/2602.11767. Version v3, 19 May 2026; method acronym TSR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Sheryl Hsu, Omar Khattab, Chelsea Finn, and Archit Sharma. Grounding by trying: LLMs with reinforcement learning-enhanced retrieval. In *International Conference on Learning Representations*, pp. 26136–26152, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4169f1425b80c3a6e7daa84a1d09b077-Paper-Conference.pdf.

Abhinav Java, Srivathsan Koundinyan, Nagarajan Natarajan, and Amit Sharma. FrugalRAG: Less is more in RL finetuning for multi-hop question answering. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ede6f43d254731152970009c172d5561-Paper-Conference.pdf.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. s3: You don't need that much data to train a search agent via RL. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21599–21617, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1095. URL https://aclanthology.org/2025.emnlp-main.1095/.

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. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516. Version v5, 5 August 2025.

Minseo Kim, Minjae Lee, Seunghyuk Oh, Kevin Galim, Donghoon Kim, Coleman Hooper, Harman Singh, Amir Gholami, Hyung Il Koo, and Wonjun Kang. EfficientRollout: System-aware self-speculative decoding for RL rollouts. *arXiv preprint arXiv:2606.18967*, 2026. URL https://arxiv.org/abs/2606.18967. Version v1, 17 June 2026.

Jiazheng Li, Yawei Wang, David Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. SALT: Step-level advantage assignment for long-horizon agents via trajectory graph. *arXiv preprint arXiv:2510.20022*, 2025a. URL https://arxiv.org/abs/2510.20022. Version v1, 22 October 2025.

Xintong Li, Sha Li, Yuwei Zhang, Changlong Yu, Rongmei Lin, Hongye Jin, Shuyi Guan, Xin Liu, Linwei Li, Qingyu Yin, and Jingbo Shang. Process reward informed tree rollout for effective multi-turn RL. *arXiv preprint arXiv:2607.15610*, 2026. URL https://arxiv.org/abs/2607.15610. Version v1, 17 July 2026; method acronym PATR.

Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. R3-RAG: Learning step-by-step reasoning and retrieval for LLMs via reinforcement learning. *arXiv preprint arXiv:2505.23794*, 2025b. URL https://arxiv.org/abs/2505.23794. Version v2, 24 October 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison 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*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Yuchen Li, Daiting Shi, Dawei Yin, and Zhicheng Dou. Agentic-R: Learning to retrieve for agentic search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15987–16005, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.785. URL https://aclanthology.org/2026.findings-acl.785/.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=ooROvpmxMV.

Kun Luo. Wiki-25-512. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/Lk123/wiki-25-512. Revision fc1e312568b14385c04f41bc09157d8fa4c20658.

Kun Luo, Hongjin Qian, Zheng Liu, Ziyi Xia, Shitao Xiao, Zhao Cao, Siqi Bao, Jun Zhao, and Kang Liu. Reinforcing agentic search via reward density optimization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10261–10283, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.467. URL https://aclanthology.org/2026.acl-long.467/.

Shichao Ma, Zhiyuan Ma, Ming Yang, Xiaofan Li, Xing Wu, Jintao Du, Yu Cheng, Weiqiang Wang, Qiliang Liu, Zhengyang Zhou, and Yang Wang. TSPO: Breaking the double homogenization dilemma in multi-turn search policy optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28607–28623, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1428. URL https://aclanthology.org/2026.findings-acl.1428/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999. URL https://people.eecs.berkeley.edu/~russell/papers/icml99-shaping.pdf.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *International Conference on Learning Representations*, pp. 60808–60838, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/98711dea460bdefe0e651ca23ec98ba2-Paper-Conference.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300. Version v3, 27 April 2024.

Jingbo Sun, Wenyue Chong, Songjun Tu, Qichao Zhang, Yaocheng Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. AutoSearch: Adaptive search depth for efficient agentic RAG via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28059–28079, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1399. URL https://aclanthology.org/2026.findings-acl.1399/.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon LLM agents. *arXiv preprint arXiv:2603.08754*, 2026. URL https://arxiv.org/abs/2603.08754. Version v1, 7 March 2026; method acronym HCAPO.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. URL https://arxiv.org/abs/2211.14275.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qkWP6phrvZ.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. SPA-RL: Reinforcing LLM agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025a. URL https://arxiv.org/abs/2505.20732. Version v1, 27 May 2025.

Yiding Wang, Zhepei Wei, Xinyu Zhu, and Yu Meng. Beyond outcome reward: Decoupling search and answering in reinforcement learning for LLM search agents. *arXiv preprint arXiv:2510.04695*, 2025b. URL https://arxiv.org/abs/2510.04695. Version v1, 6 October 2025; method acronym DeSA.

Haoze Wu, Chuqiao Kuang, Tianyi Zhuang, and Xiaoguang Li. Beyond outcome rewards: Step-level self-distilled policy optimization for deep search agents. *arXiv preprint arXiv:2608.12764*, 2026a. URL https://arxiv.org/abs/2608.12764. Version v1, 13 August 2026; method acronym SSPO.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. HiPRAG: Hierarchical process rewards for efficient agentic retrieval augmented generation. In *International Conference on Learning Representations*, pp. 82317–82347, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/85255cf8f2d2318cc7cffd0b1bc66625-Paper-Conference.pdf.

Ziyi Xia, Kun Luo, Hongjin Qian, and Zheng Liu. Open data synthesis for deep research. *arXiv preprint arXiv:2509.00375*, 2025. URL https://arxiv.org/abs/2509.00375.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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

Junkeun Yi, Damon Mosk-Aoyama, Baihe Huang, Ritu Gala, Charles Wang, Sugam Dipak Devare, Khushi Bhardwaj, Abhibha Gupta, Oleksii Kuchaiev, Jiantao Jiao, Jian Zhang, and Venkat Srinivasan. PivotRL: High accuracy agentic post-training at low compute cost. *arXiv preprint arXiv:2603.21383*, 2026. URL https://arxiv.org/abs/2603.21383. Version v1, 22 March 2026.

Yaocheng Zhang, Haohuan Huang, Zijun Song, Yuanheng Zhu, Qichao Zhang, Zijie Zhao, and Dongbin Zhao. CriticSearch: Fine-grained credit assignment for search agents via a retrospective critic. *arXiv preprint arXiv:2511.12159*, 2025. URL https://arxiv.org/abs/2511.12159. Version v1, 15 November 2025.

Yijun Zhang, Fan Xu, Jiaxin Ding, Yule Xie, Shiqing Gao, Xin Ding, Haoxiang Zhang, Luoyi Fu, and Xinbing Wang. Information gain-based rollout policy optimization: An adaptive tree-structured rollout approach for multi-turn LLM agents. *arXiv preprint arXiv:2607.06223*, 2026. URL https://arxiv.org/abs/2607.06223. Version v1, 7 July 2026; method acronym IGRPO.

Yida Zhao, Kuan Li, Xixi Wu, Liwen Zhang, Ding-Chu Zhang, Baixuan Li, Maojia Song, Zhuo Chen, Chenxi Wang, Xinyu Wang, Kewei Tu, Pengjun Xie, Fei Huang, Jingren Zhou, and Yong Jiang. Repurposing synthetic data for fine-grained search agent supervision. In *International Conference on Learning Representations*, pp. 14467–14492, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1801a817e29e8e69711dd7935651691c-Paper-Conference.pdf.

Yubao Zhao, Weiquan Huang, Sudong Wang, Ruochen Zhao, Chen Chen, Yao Shu, and Chengwei Qin. BranPO: Scalable contrastive branch sampling for long-horizon agentic reinforcement learning. *arXiv preprint arXiv:2602.03719*, 2026b. URL https://arxiv.org/abs/2602.03719. Version v2, 1 June 2026.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

Yuqi Zhou, Sunhao Dai, Changle Qu, Liang Pang, Jun Xu, and Ji-Rong Wen. Learning to retrieve from agent trajectories. *arXiv preprint arXiv:2604.04949*, 2026. URL https://arxiv.org/abs/2604.04949. Accepted at SIGIR 2026.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. AT^{2}PO: Agentic turn-based policy optimization via tree search. *arXiv preprint arXiv:2601.04767*, 2026. URL https://arxiv.org/abs/2601.04767. Version v1, 8 January 2026.

A Training Procedure

Each training round consists of 20 ABF optimizer steps followed by 20 full-trajectory Outcome steps. The current policy refreshes the frontier pool at the start of each ABF block and after its first ten updates. Each ABF step draws distinct frontier states.

We measure answer-bearing coverage on held-out validation questions after each complete round. After two consecutive rounds without improvement, we stop scheduling ABF blocks and assign the remaining budget to Outcome updates.

For the frontier in Equation 3, let $k_F = t_F + 1$ denote the one-indexed target interaction. Under a forced-answer horizon of N tool interactions, we exclude candidates with $k_F \geq N$.

Frontier depths 0, 1, 2–3, and at least 4 are sampled in a 2:2:3:1 ratio.

Frontier eligibility and coverage. Frontier eligibility does not require a correct final answer: an unsuccessful trajectory can still provide an eligible frontier if a tool observation contains an answer match. Selection therefore conditions on observed lexical acquisition, rather than successful task completion. Pool refresh updates the available frontiers as the policy changes, while depth stratification preserves deeper opportunities among those already discovered. Neither guarantees discovery of previously unreached frontiers. The depth ablation supports stratification as a pool-construction choice, but does not isolate the marginal contribution of individual frontier depths.

B Data, Retrieval, and Agent Interface

B.1 Data Roles

Training, validation, and evaluation questions are disjoint. Table 4 lists the source pool for Outcome training and frontier mining, the validation set, and the evaluation sets. Validation questions are sampled randomly within each source.

Table 4: Data sources, splits, and sizes. Evaluation subsets are fixed across methods and model scales.

Use / dataset	Source and selection	Questions
Training pool	Three training splits; named-entity answers, 3,500 per source	10,500
Validation	Held-out questions from the three training sources; 70 per source	210
Evaluation sets		
HotpotQA	Fixed subset of the full-wiki development split	830
2WikiMultiHopQA	Development subset from R1-Searcher	500
MuSiQue	Development subset from R1-Searcher	500
InfoSeek	Fixed InfoSeek Eval snapshot (text question answering)	300
Bamboogle	Complete official release	125
PopQA	Seed-42 sample from the official test split	500

Evaluation data and model input. The PopQA sample is fixed before model evaluation. The policy's initial input is the question text; reference passages and supporting-fact annotations are not supplied as context.

B.2 Retrieval System

Search returns ten candidates with at most 256 characters per snippet, and `get_document` returns at most 3,000 characters. The model context is 16,384 tokens, with at most 15,000 prompt tokens and 1,024 generated tokens per interaction.

B.3 Agent Prompt and Tool Schemas

The system prompt is:

> You are a retrieval agent. Answer the user's factual question using the available tools. Use `search` to find relevant documents. Use `get_document` only with a `docid` returned by an earlier search. Rewrite queries when the current evidence is insufficient. When the evidence supports one answer, stop using tools and return exactly one short line: `Final Answer: <answer>`. Call at most one tool per turn. The final answer must be the shortest answer entity or phrase, without an explanation, restatement, Markdown, or supporting sentence. For a Boolean question, use only `yes` or `no` after the prefix. Do not expose tool syntax, hidden evaluator data, or internal reasoning in the final answer.

The `search` schema accepts one required string field, `query`. The `get_document` schema accepts one required string field, `docid`, which must identify a document returned by an earlier search. Both schemas reject additional fields, parallel calls are disabled, and the agent may issue at most one tool call per turn.

C Rewards and Optimization

C.1 Baseline Adaptations

We adapt the released implementations to the shared `search`/`get_document` interface and use normalized EM for terminal-answer correctness. All four baselines train on complete trajectories; the method-specific adaptations are summarized below.

Search-R1. We use the same terminal EM and answer-format scoring as our Outcome updates, with a single group-relative advantage assigned across each trajectory.

DeSA. We retain the recall-then-Outcome schedule. The first stage uses a binary answer-recall reward over the accumulated outputs of both retrieval tools; the second stage switches to terminal EM. Each stage uses its own trajectory-level reward.

TSPO. We use the all-wrong-group variant: when all answers in a rollout group are incorrect, turns up to and including the first answer occurrence receive partial credit ($\alpha = 1$), and later turns receive zero. Trajectories with no occurrence receive zero throughout. These rewards are normalized within the question group at each turn and applied to that turn's sampled tokens; other groups retain ordinary Outcome supervision.

IGPO. We retain turn-wise information gain, computed from changes in the reference answer's mean per-token log probability across a tool interaction, and replace the original terminal F1 criterion with EM. Information-gain and terminal rewards are normalized separately within each question group, then combined with unit weights into undiscounted turn returns ($\gamma = 1$). Both signals remain active throughout training, without a curriculum. Likelihood probes are evaluated separately and do not alter the policy's visible history.

C.2 Answer Matching

The evaluator applies Unicode NFKC normalization, case folding, punctuation and English-article removal, and whitespace collapsing. After normalization, final-answer EM requires the predicted answer to exactly equal the canonical answer or a dataset-provided alias. The answer-bearing predicate checks for occurrences of the canonical answer or a retained alias at token boundaries within normalized tool text. For this predicate, we retain the canonical answer but remove optional aliases shorter than three alphanumeric characters or colliding with common function words. Search results are deduplicated by document identifier before answer-bearing documents are counted.

Lexical matching can reward unsupported mentions or ambiguous answer strings, and can miss paraphrases or other semantically sufficient text without a retained answer form. These limitations apply to frontier selection, local acquisition rewards, and bearing diagnostics; the predicate does not establish semantic support.

Table 2 and Figure 3 quantify acquisition under the same lexical predicate used by ABF; they do not independently validate semantic evidence quality. Final-answer improvements support downstream task effectiveness, but do not establish that every additional lexical match provides supporting evidence.

C.3 Reward and Optimization Settings

ABF combines the action-specific rewards in Section 3.3 with a format bonus $\beta_{\text{fmt}} = 0.02$ and a penalty for invalid outputs:

$$R_{\text{ABF}} = \begin{cases} R_{\text{search}}(o; y) + \beta_{\text{fmt}}, & \text{if it calls } \texttt{search(query)}, \\ R_{\text{doc}}(o; y) + \beta_{\text{fmt}}, & \text{if it calls } \texttt{get_document(docid)}, \\ R_{\text{term}}(\hat{y}; y) + \beta_{\text{fmt}}, & \text{if it is a parsed terminal response in the required format,} \\ -1, & \text{otherwise.} \end{cases} \tag{7}$$

For a valid terminal response, Outcome training uses $\text{EM}(\hat{y}, y) + \beta_{\text{fmt}}$; malformed or illegal actions receive -1. Equation 2 specifies the binary task component, excluding format scoring.

All methods exclude retrieved observations from the policy loss. Table 5 summarizes the optimization settings, including batch and group sizes.

Table 5: Shared and phase-specific optimization settings. The Outcome column also specifies the full-trajectory rollout batch used by the learned baselines.

Setting	Outcome update	ABF update
Questions or states per step	4 questions	8 frontier states
Continuations per question or state	16	8
Generation batch size	64	64
Sampling	$T = 1.0$, top-$p = 0.9$, top-$k = 50$	same
Rollout horizon	complete, at most 10 tools	one decision
Learning rate	10^{-5}, constant schedule	
GRPO clipping	$\epsilon_{\text{low}} = 0.20$, $\epsilon_{\text{high}} = 0.28$	
KL coefficient	0.001	
LoRA	rank 8, alpha 32; `o_proj`, `down_proj`, `out_proj`	

Experiments run on NVIDIA A800 80GB GPUs.

Reproducibility and training variability. For the main results in Table 1, ABF and Search-R1, the Outcome-only reference, each report the mean of three training runs with different random seeds at both model scales. DeSA, TSPO, and IGPO each use one training run per scale because of training cost. For each model scale and seed, we independently create Base with 20 Outcome warm-up updates; methods using that seed share this Base. Each training run then receives an independent budget of at most 300 optimizer steps and selects its own checkpoint by the same validation-EM rule. Training-question order is determined by the training seed and held fixed within a run; the data splits and validation set are shared across seeds. For ABF and Search-R1 in the main table, we average scores from independently selected checkpoints, rather than selecting the best run. Repeated training supports the direct comparison with Outcome-only training; the other baselines broaden coverage across supervision approaches, but their across-run variability is not measured.

D Qualitative Search Examples

Table 6 illustrates two changes in ABF's retrieval strategy: fuller initial queries for first-hop recall, and more targeted follow-up queries as intermediate entities become available.

Case 1: Fuller initial queries (HotpotQA)
Question: Erwin Lambert was a member of which paramilitary organization under Hitler?
Reference answer: Schutzstaffel

Search-R1	**ABF Training**
Erwin Lambert paramilitary organization Hitler	*Erwin Lambert member paramilitary organization under Hitler*
Bearing: 0/10 AB	**Bearing:** 3/10 AB
Final-answer EM: 1	**Final-answer EM:** 1

Case 2: More targeted follow-up queries (MuSiQue)
Question: What is the name of the castle in the city where the headquarters of the production company of Miss Spider's Sunny Patch Kids is located?
Reference answer: Casa Loma

Search-R1	**ABF Training**
Miss Spider's Sunny Patch Kids production company headquarters	*Nelvana headquarters Toronto* (0/10 AB)
→ *Nelvana headquarters city*	→ *Toronto castle* (1/10 AB)
	→ *Casa Loma Toronto castle* (8/10 AB)
Final answer: Royal Castle	**Final answer:** Casa Loma
Final-answer EM: 0	**Final-answer EM:** 1

Table 6: **Qualitative search comparisons.** Selected queries and final outcomes from evaluation trajectories. Arrows connect the selected queries; AB counts distinct answer-bearing documents among ten search results.

ABF uses fuller initial queries to improve first-hop retrieval. ABF retains more words from the question when formulating its first search, giving the retriever a fuller description of the information need. In the HotpotQA example, its query adds "member" and "under" to Search-R1's shorter

query and returns three answer-bearing documents, versus none for Search-R1. Across evaluation questions, mean first-query length increases from 7.1 to 11.4 words. This is consistent with the higher question-term retention and first-search hit rate in Table 2. Both policies ultimately answer correctly in this example.

ABF makes follow-up queries more targeted. Both policies search for Nelvana's headquarters. Search-R1's displayed query asks for the headquarters "city"; ABF names Toronto and carries it into "Toronto castle", turning the intermediate city into a concrete constraint for the next hop. It then combines "Casa Loma" with "Toronto" to retrieve information about a specific candidate. This sequence illustrates how ABF updates its retrieval target using intermediate findings: from the company's headquarters, to a castle in that city, and then to a named castle.

E Evaluation Protocol and Cost Accounting

Final-answer evaluation. After hidden reasoning blocks are removed, the evaluator parses exactly one `Final Answer:` marker from the final non-empty line and assigns zero to malformed responses or prompt overflows. Answer normalization and alias matching follow Appendix C.2.

Pool-construction ablation. Figure 2(b) evaluates direct and depth-aware pool construction after 20 ABF and 40 Outcome updates from the shared Base. The stated gains use the one-decimal displayed scores (2B: $32.4 - 23.3 = 9.1$; 9B: $46.2 - 40.5 = 5.7$).

Sampling-cost accounting. We audit the first 120 ABF updates and the paired 120 Outcome updates on one recorded Qwen3.5-2B training path, after the common Outcome-20 Base. These Outcome blocks belong to the alternating ABF training path; their average cost is not the per-step cost of the standalone Search-R1 arm. For model requests j, sampling tokens are $T_{\text{sample}} = \sum_j (T_{\text{prompt},j} + T_{\text{completion},j})$. Repeated history inputs are counted at every request. Tool text contributes when included in subsequent prompts, but retrieval and embedding computation are not measured by this token count.

Mining-inclusive sampling costs. Table 7 separates the selected-prefix lower bound used in Figure 5, a fixed-pool early-stopping counterfactual, and recorded execution including complete mining trajectories. All three retain eight newly sampled one-decision continuations per state; none assumes reuse of the mining witness as a training continuation. Across 120 updates, local continuations cost 20,408,644 tokens. The respective mining totals are 4,279,124, 70,690,197, and 160,673,776 tokens; paired Outcome sampling totals 204,759,436 tokens.

Table 7: Sampling tokens per optimizer step on the audited 2B path (millions). Ratios compare sampling only. Early stopping is a counterfactual replay estimate; the Outcome reference comprises blocks on the same ABF path.

Accounting	Mining	Local	Total	vs. Outcome
Selected-prefix lower bound	0.036	0.170	0.206	12.1%
Fixed-pool early stopping (estimate)	0.589	0.170	0.759	44.5%
Recorded full mining	1.339	0.170	1.509	88.4%
Paired Outcome sampling	—	—	1.706	100%

Candidate yield and training selection. The 12 pool-construction blocks, including replenishment sampling, contain 7,872 mining trajectories. Of these, 4,807 (61.1%) provide a materialized legal frontier, and 960 states are selected for training: 12.2% of mining trajectories or 20.0% of legal candidates. Thus, each selected state entails 8.2 mining trajectories on average. Candidate yield and training selection are different quantities; domain and depth quotas can leave legal candidates unused. These are trajectory and state-selection counts across refreshes, not counts of distinct questions. The eight local continuations per state do not multiply the source-state count.

Early-stopping counterfactual. Replay preserves the historical candidate pools, replenishment, quotas, selected states, and update counts. For a trajectory with a materialized legal frontier, requests are counted through the witness action and later requests are omitted. Otherwise, counting stops at the original trajectory end or the ninth request, whichever comes first, consistent with the legal frontier requirement $k_F < 10$. Under this rule, selected frontier prefixes, unselected legal-frontier prefixes, and trajectories without a materialized legal frontier cost 4,279,124, 8,193,019, and 58,218,054 tokens, respectively. The last category accounts for 82.4% of the replayed mining cost. The 0.759M estimate therefore retains unsuccessful mining and selection overhead; it does not assume perfect frontier discovery. This fixed-trajectory replay is not a measured early-stopping implementation or a new training run, and does not establish the effect of altered scheduling or smaller pools on accuracy.

Per-update work proxy and the one-fifth estimate. To distinguish sampling from optimization work, we also report the proxy $W = T_{\text{sample}} + 3T_{\text{update}}$, where T_{update} counts tokens in sequences used for GRPO updates. The factor three is a fixed approximation for forward/backward update work; W is expressed in token-equivalents, not measured FLOPs or elapsed time. Across the same 120 updates, ABF and paired Outcome update sequences contain 11,871,427 and 36,076,239 tokens, respectively. With selected-prefix sampling, ABF therefore uses $(24,687,768 + 3 \times 11,871,427)/120 \approx 0.502517$M token-equivalents per step, versus $(204,759,436 + 3 \times 36,076,239)/120 \approx 2.608235$M for paired Outcome. The ratio is 19.3%, giving the "about one fifth" estimate. This lower bound excludes unselected mining and source suffixes; it is distinct from Figure 5's sampling-only ratio of 12.1%. Keeping the update term fixed, the early-stopping counterfactual and full recorded mining give 1.055943M (40.5%) and 1.805806M (69.2%) token-equivalents per step. The latter accounts for all recorded mining in the audited lineage. These proxies do not measure hardware utilization, caching, retrieval computation, or end-to-end training efficiency.

Table 8: How the reported cost ratios relate. All entries compare average ABF and paired Outcome updates on the same audited 2B path. Sampling uses T_{sample}; work uses $W = T_{\text{sample}} + 3T_{\text{update}}$.

Mining accounting	Sampling ratio	Work ratio
Selected-prefix lower bound	12.1%	19.3% ($\approx$ one fifth)
Early stopping (counterfactual)	44.5%	40.5%
Full recorded mining	88.4%	69.2%

Table 8 separates two choices: the metric being compared (columns) and how much mining is included (rows). The 12.1% sampling ratio and the 19.3% work ratio both use the selected-prefix lower bound; neither is a full-mining cost ratio. The sampling and work denominators are 1.706329M tokens and 2.608235M token-equivalents per paired Outcome step, respectively. These ratios are not interchangeable.

Scope of the cost claims. The recorded-execution total includes all recorded mining requests in the audited training lineage, including unselected trajectories and source suffixes. Sampling totals still exclude optimizer work, common initialization, validation/evaluation, failed branches outside that lineage, retrieval and embedding computation, and system overhead. They are neither measured FLOPs nor wall-clock time, GPU-hours, or monetary costs. Figure 5 retains the selected-prefix accounting for both its curve and step breakdown; it should not be interpreted as a comparison at matched full execution cost. The main table instead compares a fixed optimizer-update budget. Neither comparison establishes end-to-end compute matching across all methods, and lower average ABF step cost does not by itself establish lower total training cost to reach a given answer accuracy.