CONFIDENCE-GUIDED QUERY REFINEMENT FOR RELIABLE SEARCH-AUGMENTED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Search-augmented reasoning enables language-model agents to acquire missing knowledge, but retrieval can fail when an agent generates a query that poorly expresses its current information need. We study this failure mode, which we term *query unreliability*, and show that token-level query confidence provides a pre-retrieval signal of retrieval quality and downstream answer correctness. Based on this observation, we introduce **Confidence-Guided Query Refinement (QC→QR)**, a post-hoc test-time method that executes confident queries unchanged while selectively rewriting, decomposing, or disambiguating low-confidence queries using an off-the-shelf reformulator. QC→QR requires no additional training of the search agent or retriever. Across 4 heterogeneous search agents and 4 knowledge-intensive QA benchmarks, QC→QR improves average answer accuracy by up to up to 6.18 percentage points, improves gold-document retrieval and downstream calibration, and requires a comparable number of external search calls. Unconditional and random refinement do not yield consistent gains, highlighting the importance of selectively intervening on uncertain queries. Our results establish query confidence as a practical signal for repairing unreliable search actions before retrieval.

1 INTRODUCTION

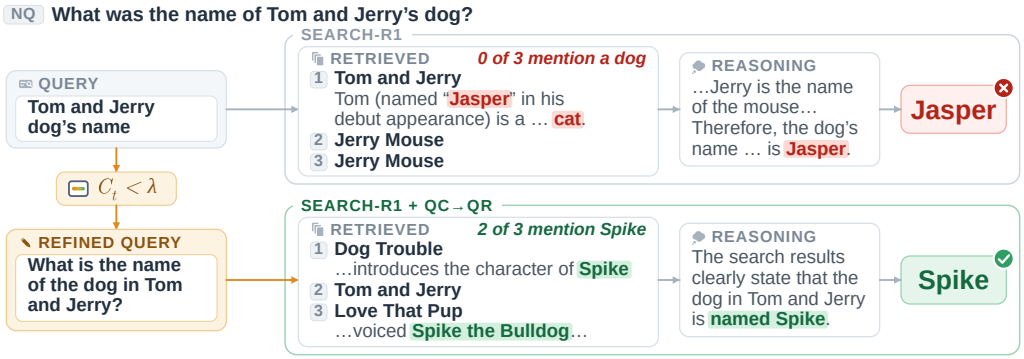
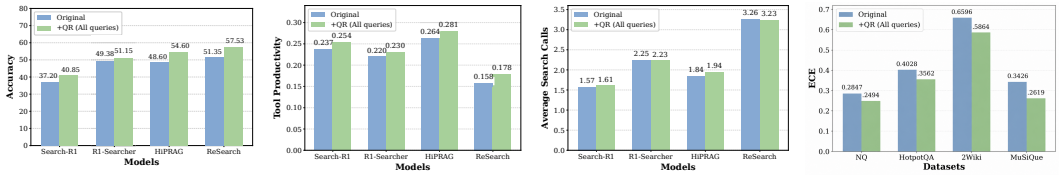


Figure 1: **Query unreliability can misdirect retrieval and downstream reasoning.** Search-R1 issues the underspecified query "Tom and Jerry dog's name," which retrieves no evidence about the dog and leads to the incorrect answer "Jasper." Because the query has low confidence ($C_t < \lambda$), QC→QR repairs it before retrieval; the refined query retrieves evidence about Spike and yields the correct answer. The example illustrates our target failure mode: even when an agent correctly decides to search, an unreliable query can prevent the needed evidence from being retrieved. Additional examples are given in Sec H (App.).

Large Reasoning Models (LRMs) perform strongly on complex reasoning tasks, but their parametric knowledge can be incomplete or outdated (OpenAI et al., 2026; Team et al., 2025). Search-augmented reasoning mitigates this limitation through external retrieval (Jin et al., 2025; Song et al., 2025b). Recent work has mainly improved *when* and *how often* agents search, typically using reinforcement learning on final-answer correctness. Yet the queries themselves receive little direct supervision and

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.

are shaped only indirectly by downstream rewards, so they may fail to express the agent's current information need.



(a) Answer accuracy. (b) Tool productivity. (c) Avg. search calls. (d) ECE

Figure 2: **Effect of confidence-guided query refinement across search agents.** Query Refinement (QR) improves (a) answer accuracy, (b) tool productivity, while (c) requiring a comparable number of search calls. It also improves calibration across datasets (d). These results indicate that confidence-guided refinement improves the accuracy, reliability, and utility of search interactions without relying on additional retrieval.

As a result, agents may issue queries that fail to express the information need implied by their current reasoning state. We call this failure *query unreliability* (Fig. 1; Sec. H in the appendix). Such queries may be ambiguous, underspecified, compositionally inappropriate, or focused on the wrong entity or relation. Because retrieval can only act on the query it receives, these errors occur upstream of evidence acquisition, yielding incomplete, irrelevant, or misdirected passages that can derail subsequent reasoning. Across the four agents, around 30% of issued search queries retrieve no gold supporting passage among the top-three results on HotpotQA, 2WikiMultiHopQA, and MuSiQue.

One approach is to improve search queries during training. SmartSearch (Wen et al., 2026) uses query-level process rewards to refine low-quality queries, while related methods reward informative search steps (Wang et al., 2025b; 2026). Search Wisely (Wu et al., 2025b) uses the minimum token probability over search queries as a rollout-level confidence signal for training. These approaches improve query generation by modifying the agent itself and require retraining for each agent.

At inference time, FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024) use token uncertainty to decide when to retrieve and to construct queries from tentative model outputs. Search agents, however, produce explicit queries as actions. Existing methods do not ask whether an already-generated search query should be executed as written or repaired before retrieval. This leads to our central question: *once an agent decides to search, can its own uncertainty identify an unreliable query before execution, and can selectively repairing such queries improve downstream reasoning?*

We find that query confidence provides a useful pre-retrieval signal of retrieval quality. Across Search-R1 and ReSearch on three benchmarks, gold-document Hit@1 generally increases with minimum query-token confidence (Fig. 3). Consistent with Wu et al. (2025b), higher confidence is also associated with greater final-answer accuracy. Because this signal is available before retrieval, it suggests a simple intervention: repair a generated query only when its confidence is low.

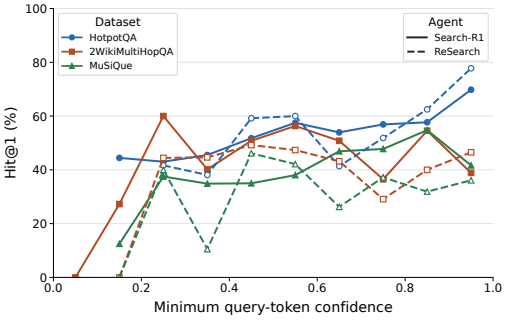


Figure 3: **Query confidence and retrieval quality.** Gold-document Hit@1 as a function of minimum query-token confidence for Search-R1 and ReSearch across HotpotQA, 2WikiMultiHopQA, and MuSiQue. Colors denote datasets, while solid and dashed lines denote the two agents. Despite variation across confidence bins, higher-confidence queries generally achieve better retrieval quality.

Building on this observation, we introduce **Confidence-Guided Query Refinement (QC→QR)**, a post-hoc test-time method that selectively refines low-confidence queries while executing confident ones unchanged. At each search step, uncertain queries are rewritten, decomposed, or disambiguated using an off-the-shelf reformulation model (Chan et al., 2024); neither the search agent nor the retriever is retrained.

Alternative refinement strategies do not match QC→QR. Rejection sampling lowers accuracy for all four agents, and matched-budget random refinement degrades all four by 2.1–8.9 percentage

points. Refining every query lowers accuracy for three of four agents and gains only 1.4 points on ReSearch, against 6.2 for QC→QR. In contrast, QC→QR improves all four agents and is the most accurate method on each. It also improves HiPRAG (Wu et al., 2026), which is already trained with process rewards, by 6.00 points, suggesting that test-time query repair can complement training-based search supervision.

Contributions. We make three main contributions: **(i) Query confidence predicts retrieval quality before search.** We show that an agent's token-level confidence in a generated query is informative of supporting-evidence retrieval across two search agents and three multi-hop benchmarks. While prior work connected this signal to final-answer correctness and used it for training (Wu et al., 2025b), we establish its utility at the individual-query level *before* retrieval. **(ii) Test-time confidence-guided query refinement.** We introduce Confidence-Guided Query Refinement (QC→QR), which selectively refines low-confidence queries while executing confident ones unchanged, requiring no additional training of the search agent or retriever. **(iii) Consistent gains from selective intervention.** Across Search-R1, R1-Searcher, HiPRAG, and ReSearch, QC→QR improves average answer accuracy by **+3.65**, **+1.78**, **+6.00**, and **+6.18** percentage points, respectively, with a comparable number of external search calls (Fig. 2). Unlike unconditional refinement, rejection sampling, and matched-budget random refinement, QC→QR improves all four agents and is the most accurate method on each. It also improves Search-R1 and R1-Searcher on open-web evaluation with GAIA and WebWalkerQA, by **2.86** and **3.65** percentage points.

2 Related Work

Training and improving search agents. Search agents are commonly trained with reinforcement learning to decide *when* and *how often* to retrieve (Jin et al., 2025; Song et al., 2025a; Chen et al., 2025). More recent work introduces process-level rewards to encourage informative and efficient search behavior (Wang et al., 2025b; 2026; Wu et al., 2026). These methods modify the agent during training, whereas QC→QR operates post hoc on an already-trained search policy.

Query refinement and search reliability. Query rewriting has long been used to improve retrieval (Chan et al., 2024; Mao et al., 2024). For search agents, SmartSearch (Wen et al., 2026, contemporaneous) uses query-level process rewards to improve query generation during training, while WebFilter (Dai et al., 2026) jointly optimizes query formulation and web retrieval. QC→QR instead leaves the search agent unchanged and selectively refines only low-confidence queries at test time using an off-the-shelf reformulator.

Uncertainty for retrieval and tool use. FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024) use token uncertainty to decide when to retrieve and how to construct queries from tentative model outputs, while Search Wisely (Wu et al., 2025b) uses minimum query-token confidence as a training signal. QC→QR instead evaluates an explicit query already produced by a trained search agent and uses its token probabilities to decide whether that query should be executed or repaired before retrieval.

A more detailed discussion of search-agent training, failure modes, query rewriting, and uncertainty-based retrieval appears in Appendix D.

3 Methodology

We introduce *Confidence-Guided Query Refinement (QC→QR)*(Query Confidence → Query Refinement), a training-free test-time intervention for search-augmented reasoning. Before executing each query, QC→QR uses the agent's own query confidence to decide whether to trust it: confident queries are executed unchanged, while uncertain ones are refined before retrieval. At each search step, the method makes two decisions: *when* to intervene (Sec. 3.2) and *how* to repair an unreliable query (Sec. 3.3). We first show where these decisions enter a standard search agent (Sec. 3.1). Figure 4 provides an overview, and Algorithm 1 gives the full procedure.

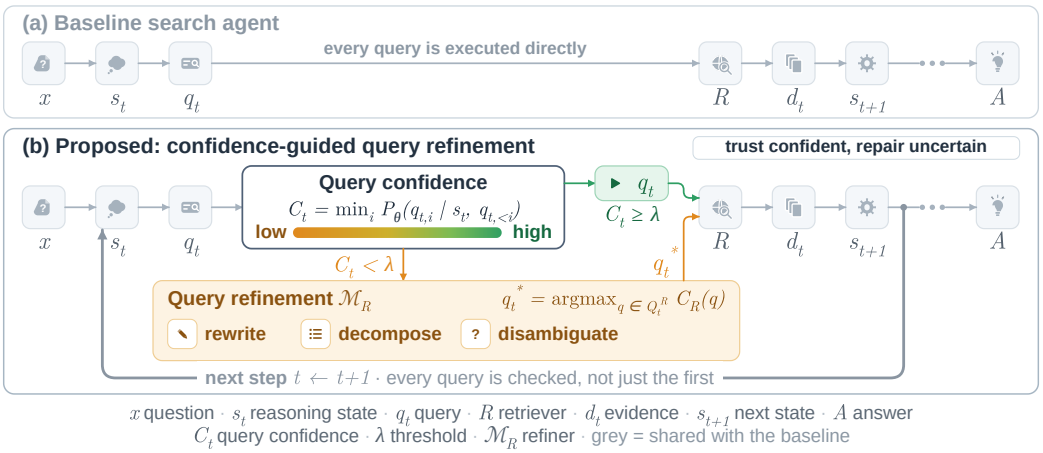


Figure 4: **Overview of QC→QR.** (a) A standard search agent executes each generated query q_t directly. (b) QC→QR computes query confidence C_t before retrieval. Queries with $C_t \geq \lambda$ are executed unchanged (green path), while queries with $C_t < \lambda$ are rewritten, decomposed, or disambiguated by the reformulation model $\mathcal{M}_R$. The candidate with the highest reformulator confidence, q_t^*, is then executed (orange path). The confidence check is applied at every search step.

3.1 Problem Setup

Consider a standard search agent given an input question x. The agent alternates between reasoning and external retrieval until producing a final answer. At search step t, its state s_t contains the question and the reasoning–retrieval history. The search policy π_θ generates a query

$$q_t \sim \pi_\theta(\cdot \mid s_t), \tag{1}$$

which is submitted to a retriever $\mathcal{R}$ to obtain

$$d_t = \mathcal{R}(q_t). \tag{2}$$

The agent then updates its state as

$$s_{t+1} = f_\theta(s_t, q_t, d_t). \tag{3}$$

We keep both π_θ and $\mathcal{R}$ fixed. QC→QR therefore operates entirely at inference time, intervening only on the query executed by the retriever. Whereas the baseline submits every generated query directly, QC→QR inserts a reliability check between query generation (Eq. 1) and retrieval (Eq. 2).

3.2 When to Refine: Query Confidence

The first decision is whether a generated query q_t should be refined before retrieval. We seek a signal that is available *before* retrieval, incurs little additional cost, and is sensitive to localized uncertainty: a query may be mostly well formed yet contain an uncertain entity, relation, or constraint that changes what is retrieved. We therefore use the search policy's own token-level generation probabilities.

Writing $q_t = (q_{t,1}, \ldots, q_{t,n_t})$, we define query confidence as the minimum probability assigned to any generated token:

$$C_t \equiv C(q_t) = \min_{1 \leq i \leq n_t} P_\theta\left(q_{t,i} \mid s_t, q_{t,<i}\right). \tag{4}$$

This minimum acts as a *weakest-link* score: a single low-confidence generation decision lowers C_t, whereas averaging token probabilities can dilute such localized uncertainty. We compare alternative confidence aggregations in Appendix G.5. Since these probabilities are already produced during query generation, computing C_t requires no auxiliary verifier or additional model call.

Given a validation-selected threshold λ, QC→QR executes

$$q_t^{\text{exec}} = \begin{cases} q_t, & C_t \geq \lambda, \\ q_t^\star, & C_t < \lambda, \end{cases} \tag{5}$$

where $q_t^\star$ is the refined query defined in Sec. 3.3. Increasing λ routes more queries to refinement, trading additional reformulation cost for more aggressive intervention.

3.3 How to Refine: Candidate Generation and Selection

When equation 5 triggers refinement, QC→QR must generate plausible repairs and select one without additional retrieval. We use three complementary operations adapted from RQ-RAG (Chan et al., 2024): *rewrite*, which rephrases the query for clarity; *decompose*, which isolates an intermediate sub-question from a composite information need; and *disambiguate*, which makes the intended entity or relation explicit. These operations target common query failures such as missing intermediate entities, ambiguity, and incorrect relation commitments (App H).

Let $\mathcal{G} = \{g_{\text{rew}}, g_{\text{dec}}, g_{\text{dis}}\}$ denote the corresponding refinement instructions. A reformulation model $\mathcal{M}_R$ generates one candidate per operation:

$$q_t^{(g)} \sim \mathcal{M}_R\left(\cdot \mid s_t, q_t, g\right), \qquad g \in \mathcal{G}, \tag{6}$$

yielding $\mathcal{Q}_t^R = \{q_t^{(g)} : g \in \mathcal{G}\}$. For decomposition, the candidate is the first sub-question; subsequent hops are handled by the agent in later search steps, each again subject to equation 5.

We select among candidates using the reformulation model's own generation confidence, defined analogously to equation 4:

$$C_R(q_t^{(g)}) = \min_{1 \le i \le n_t^{(g)}} P_{\mathcal{M}_R}\left(q_{t,i}^{(g)} \mid s_t, q_t, g, q_{t,<i}^{(g)}\right), \tag{7}$$

where $n_t^{(g)}$ is the candidate length. We then execute

$$q_t^\star = \arg\max_{q \in \mathcal{Q}_t^R} C_R(q). \tag{8}$$

This selection requires no retrieval or additional call to the search agent. The selected query $q_t^\star$ is submitted to the same retriever in place of q_t, after which the original reasoning loop continues unchanged.

4 Experimental Setup

Search agents. We evaluate QC→QR on four publicly released search agents: Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025a), ReSearch (Chen et al., 2025), and HiPRAG (Wu et al., 2026). The first three are trained with outcome-based rewards, whereas HiPRAG uses hierarchical process rewards. We allow up to 5 search turns, a 4096-token context, and 256 generated tokens per turn, with temperature 0.7 and top-$p = 0.95$.

QC→QR configuration. Query confidence is computed from the agent's token probabilities using equation 4. As the reformulator $\mathcal{M}_R$, we use the MR checkpoint of RQ-RAG (Chan et al., 2024), instantiated with Llama2-7B; the search agent and retriever remain fixed. For each agent, the threshold λ is selected on disjoint validation subsets of 200 questions per dataset and fixed before evaluation. The resulting thresholds are 0.4125, 0.33, 0.36, and 0.175 for Search-R1, R1-Searcher, HiPRAG, and ReSearch, respectively. We evaluate **QR (All)**, which checks every search query, and **QR (First)**, which checks only the first query in each trajectory.

Benchmarks and retrieval. We evaluate on Natural Questions (NQ) (Kwiatkowski et al., 2019) for single-hop open-domain QA, and HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2023) for multi-hop reasoning. We sample 500 questions per dataset for Search-R1 and 200 per dataset for R1-Searcher, HiPRAG, and ReSearch, following prior evaluation subsets (App. E.1). Retrieval uses the 2018 Wikipedia corpus indexed with FAISS using `intfloat/e5-base-v2` embeddings, with the top-3 passages returned per query. To test generalization beyond a fixed index, we additionally evaluate Search-R1 and R1-Searcher zero-shot on GAIA (Mialon et al., 2023) and WebWalkerQA (Wu et al., 2025a) using the Serper API and its top-10 returned snippets.

Baselines. We use two groups of baselines. As reference points, we report direct inference, Chain-of-Thought (CoT) (Wei et al., 2023), RAG (Lewis et al., 2021), IRCoT (Trivedi et al., 2023), and each search agent without intervention (*Original*). To isolate the components responsible for QC→QR's gains, we also evaluate three test-time controls. *Refine All* applies the reformulator to every query, removing the confidence gate. *Random Refinement* applies the same reformulator to randomly selected queries with an intervention budget matched to QC→QR. *Rejection Sampling* retains the confidence gate but replaces the reformulator: when $C_t < \lambda$, it resamples from the search agent for up to five attempts until $C_t \geq \lambda$, using the final sample if none passes. All controls use the same underlying agent, retriever, and evaluation questions as QC→QR.

Metrics. Answer accuracy is judged by Qwen3-8B with a fixed evaluation prompt that assesses semantic consistency with the reference answer, following Chen et al. (2025). To analyze why QC→QR works, we measure retrieval quality with Hit@n (Ammann et al., 2025), whether a gold supporting passage is among the top-n retrieved passages; repeated retrieval with $\mathrm{NovelRate@}\,K = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}[O_t \leq K]$, where O_t is the number of passages retrieved at step t that already appeared earlier in the trajectory (Wen et al., 2026); and final-answer calibration with ECE (Guo et al., 2017), ACE (Nixon et al., 2019), MCE, Brier score, and NLL, computed from the answer confidence C_{ans}, the geometric mean of the answer-token probabilities (i.e., the length-normalized sequence probability of the answer). To measure cost, we report search calls per question, tool productivity (correct answers per search call), the fraction of search queries that are refined, and runtime. For Definitions see App. E.3.

Protocol. Each configuration is evaluated once. Main experiments use seed 42; the test-time controls use seed 40 for random selection and generation. Answer accuracy and control baselines are reported for all four agents. Additional analyses use subsets of agents: query-confidence versus retrieval quality for Search-R1, ReSearch, and HiPRAG; open-web evaluation for Search-R1 and R1-Searcher; and retrieval novelty for Search-R1. Calibration results for Search-R1 are reported in the main paper, with the remaining agents in App. G.

A small number of evaluation trajectories could not be completed because of retrieval failures and endpoint timeouts. Consequently, HiPRAG on MuSiQue uses 135/200 questions, R1-Searcher on NQ uses 191/200, and ReSearch on HotpotQA uses 197/200 for answer-accuracy evaluation.

5 Results

We first evaluate whether QC→QR improves search-augmented reasoning and whether its gains depend on confidence-guided intervention (Sec. 5.1). We then analyze how query confidence, retrieval quality, and downstream outcomes are related (Sec. 5.2), before studying design choices (Sec. 5.3).

5.1 Main Results

QC→QR improves all four agents. Table 1 reports answer accuracy. QR (All) improves average accuracy by **+3.65**, **+1.78**, **+6.00**, and **+6.18** points for Search-R1, R1-Searcher, HiPRAG, and ReSearch, respectively, while improving or matching the original agent on every agent–dataset pair. The gain on HiPRAG shows that test-time query repair can complement an agent already trained with process rewards. QR (First) captures only part of these gains, indicating that useful interventions occur beyond the first search step.

The gains require both the gate and the reformulator. A natural concern is that these gains come from the added reformulator or the extra generation, rather than from deciding which queries to refine. Table 2 removes one ingredient at a time. Without the gate, Refine All lowers accuracy for Search-R1, R1-Searcher, and HiPRAG (-1.95, -1.42, -2.12) and improves ReSearch by only $+1.38$, against $+6.18$ for QC→QR; Random Refinement with the same budget lowers accuracy for all four agents, by 2.12 to 8.94 points. Without the reformulator, Rejection Sampling lowers accuracy for all four agents, by 1.00 to 11.45 points. Only the combination improves all four agents, and it is the most accurate method on each, with a comparable number of search calls.

Table 1: **QC→QR improves search-augmented reasoning.** **First** refines only the first query, whereas **All** selectively refines queries throughout the trajectory. Green values denote absolute accuracy gains over the corresponding base agent.

Method	NQ	Hotpot	2Wiki	MuSiQue	Avg.
Inference Baselines					
Direct	13.4	14.5	18.5	3.1	12.38
CoT	4.8	18.5	22.0	2.2	11.88
RAG	33.0	19.5	24.2	5.8	20.63
IRCoT	22.4	26.1	14.2	7.2	17.48
Search-Augmented Agents					
Search-R1 (Jin et al., 2025)	48.4	40.4	35.0	25.0	37.20
+QR (First)	52.8	42.6	37.6	26.8	39.95 ↑2.75
+QR (All)	**53.2**	**43.6**	**38.8**	**27.8**	**40.85** ↑**3.65**
R1-Searcher (Song et al., 2025a)	55.5	60.0	52.0	30.0	49.38
+QR (First)	56.0	61.0	52.0	31.0	50.00 ↑0.63
+QR (All)	**57.6**	**62.5**	**52.0**	**32.5**	**51.15** ↑**1.78**
HiPRAG (Wu et al., 2026)	61.5	50.0	58.5	24.4	48.60
+QR (First)	63.0	51.5	64.5	28.9	51.98 ↑3.38
+QR (All)	**65.0**	**55.0**	**66.5**	**31.9**	**54.60** ↑**6.00**
ReSearch (Chen et al., 2025)	53.0	61.9	53.0	37.5	51.35
+QR (First)	56.0	65.0	54.0	41.0	54.00 ↑2.65
+QR (All)	**59.0**	**70.1**	**57.5**	**43.5**	**57.53** ↑**6.18**

Table 2: **Accuracy and retrieval cost across test-time control methods.** ΔAcc. is relative to the corresponding original agent; Search/Q is the average number of external retrieval calls per question.

Agent	Method	Acc. (%)	ΔAcc.	Search/Q
Search-R1	Original	37.20	–	1.57
	QC→QR	**40.85**	**+3.65**	1.61
	Refine All	35.25	-1.95	1.76
	Reject. Samp.	25.75	-11.45	0.62
	Random Ref.	31.80	-5.40	1.65
R1-Searcher	Original	49.38	–	2.25
	QC→QR	**51.15**	**+1.77**	2.23
	Refine All	47.96	-1.42	2.27
	Reject. Samp.	45.57	-3.81	2.08
	Random Ref.	40.44	-8.94	2.84
HiPRAG	Original	48.60	–	1.84
	QC→QR	**54.60**	**+6.00**	1.94
	Refine All	46.48	-2.12	1.89
	Reject. Samp.	47.60	-1.00	1.92
	Random Ref.	46.48	-2.12	1.90
ReSearch	Original	51.35	–	3.26
	QC→QR	**57.53**	**+6.18**	3.23
	Refine All	52.73	+1.38	3.19
	Reject. Samp.	48.58	-2.77	3.14
	Random Ref.	47.46	-3.89	3.33

The gains extend to open-web search. To test whether the effect holds beyond a fixed Wikipedia index, Table 4 evaluates QC→QR zero-shot on GAIA and WebWalkerQA with web search. QC→QR improves average accuracy by 2.86 points for Search-R1 and 3.65 points for R1-Searcher, and raises tool productivity for both. Absolute accuracy remains low, as expected for agents trained on Wikipedia-based QA. Confidence-gated refinement therefore improves agents across settings; we next examine why it works.

5.2 Why Confidence-Guided Refinement Works

Table 3: **Effect of confidence-guided query refinement on gold-document retrieval per question.** Hit@1 and Hit@3 measure whether a gold supporting document is retrieved among the top-1 or top-3 results. *Original* denotes retrieval with the agent-generated query and *QR* retrieval after confidence-guided refinement. Parenthesized values are absolute percentage-point changes over Original.

Agent	Dataset	Hit@1 (%)		Hit@3 (%)	
		Original	QR	Original	QR
Search-R1	HotpotQA	62.80	70.20 (+7.40)	70.60	77.00 (+6.40)
	2Wiki	63.20	69.00 (+5.80)	70.80	74.20 (+3.40)
	MuSiQue	49.00	58.00 (+9.00)	60.80	67.20 (+6.40)
R1-Searcher	HotpotQA	74.00	75.00 (+1.00)	79.50	82.00 (+2.50)
	2Wiki	76.50	78.50 (+2.00)	80.50	83.50 (+3.00)
	MuSiQue	61.50	63.50 (+2.00)	68.50	71.00 (+2.50)
ReSearch	HotpotQA	79.38	83.51 (+4.12)	83.51	85.57 (+2.06)
	2Wiki	78.00	80.00 (+2.00)	81.00	82.00 (+1.00)
	MuSiQue	59.50	62.00 (+2.50)	73.50	75.00 (+1.50)
HiPRAG	HotpotQA	65.50	73.00 (+7.50)	78.00	81.50 (+3.50)
	2Wiki	63.50	69.50 (+6.00)	73.00	76.50 (+3.50)
	MuSiQue	45.19	53.33 (+8.15)	60.00	67.41 (+7.41)

We trace the effect of QC→QR through the search pipeline: whether confidence identifies weak queries, whether refinement improves their retrieval, and how these changes affect subsequent trajectories and final answers.

Table 4: **Open-web evaluation.** QC→QR improves answer accuracy (Acc.) and tool productivity (TP) on GAIA and WebWalkerQA.

Method	GAIA		WebWalker		Average	
	Acc.	TP	Acc.	TP	Acc.	TP
Search-R1	6.00	4.88	7.38	5.16	6.69	5.02
+QR	**9.00**	**8.00**	**10.09**	**7.64**	**9.55 (+2.86)**	**7.82**
R1-Searcher	4.60	5.20	5.20	1.84	4.90	3.52
+QR	**10.70**	**12.10**	**6.40**	**2.40**	**8.55 (+3.65)**	**7.25**

Low confidence marks queries that retrieve poorly. Figure 3 shows that gold-document Hit@1 generally increases with query confidence. At each agent's selected threshold λ, Hit@1 is lower for queries with $C_t < \lambda$ than for those with $C_t \geq \lambda$: 40.4% vs. 48.7% for Search-R1, 34.7% vs. 49.4% for R1-Searcher, 20.0% vs. 41.7% for HiPRAG, and 36.8% vs. 42.4% for ReSearch across HotpotQA, Musique and 2Wiki dataset. At the trajectory level, the confidence of the least-confident query, $C_\tau = \min_{1 \leq t \leq T} C_t$, achieves an AUROC of 0.631 for final-answer correctness and is positively associated with answer accuracy (App. Fig. 7). These results support using low query confidence as a pre-retrieval signal for intervention.

Refinement repairs retrieval. If low-confidence queries retrieve poorly, repairing them should improve retrieval. Table 3 compares gold-passage retrieval with the agent's original queries and with the queries executed by QC→QR Hit@1 is computed over all the queries generated from the test set by the agents. Hit@1 improves for every agent and multi-hop dataset, by 1.00 to 9.00 points, with the largest gains for Search-R1 ($+5.80$ to $+9.00$) and HiPRAG ($+6.00$ to $+8.15$); Hit@3 improves by 1.00 to 7.41 points. R1-Searcher, which triggers refinement least often, shows the smallest gains.

Table 5: **Novel retrieval rate (%) for Search-R1 before and after QC→QR.**

	$K = 1$		$K = 2$	
Dataset	**Orig.**	**QR**	**Orig.**	**QR**
NQ	72.81	**88.70**	85.38	**93.33**
2Wiki	92.86	**94.71**	96.03	**96.88**
HotpotQA	81.41	**87.72**	85.34	**90.23**
MuSiQue	88.71	**91.11**	94.35	**95.19**

Refinement also reduces redundant retrieval. For Search-R1, QC→QR increases NovelRate@1 and NovelRate@2 on all four benchmarks (Table 5), with the largest gain on NQ at $K = 1$ ($72.81\% \rightarrow 88.70\%$). Because novelty measures diversity rather than relevance, we interpret this result together with the Hit@n gains above: refinement retrieves evidence that is both more relevant and less repetitive.

Table 6: **QC→QR improves final-answer calibration for Search-R1.** Each cell reports Original → QR; lower is better and the better value is bold. Results for other agents are in App. G.

Dataset	ECE ↓	ACE ↓	MCE ↓	Brier ↓	NLL ↓
NQ	0.2847 → **0.2494**	0.2854 → **0.2516**	0.5864 → **0.3046**	0.3068 → **0.2959**	**1.1336** → 1.1525
HotpotQA	0.4028 → **0.3562**	0.4028 → **0.3562**	0.6792 → **0.5590**	0.4069 → **0.3784**	1.6100 → **1.5781**
2Wiki	0.3426 → **0.2619**	0.3379 → **0.2601**	**0.3679** → 0.4188	0.3600 → **0.2957**	1.4729 → **1.2366**
MuSiQue	0.6596 → **0.5864**	0.6599 → **0.5868**	0.7379 → **0.6591**	0.6100 → **0.5523**	2.2695 → **2.1404**
Combined	0.4004 → **0.3434**	0.4003 → **0.3437**	0.5683 → **0.3921**	0.4206 → **0.3806**	1.6215 → **1.5269**

Better retrieval improves downstream calibration. For Search-R1, QC→QR lowers combined ECE from 0.4004 to 0.3434, MCE from 0.5683 to 0.3921, Brier score from 0.4206 to 0.3806, and NLL from 1.6215 to 1.5269 (Table 6). Improvements hold for nearly all dataset–metric pairs, except NLL on NQ and MCE on 2Wiki. Answer confidence, computed after retrieval, need not reflect whether the query was reliable; repairing it can make confident answers correct more often (App. C). Having established why QC→QR works, we turn to its design choices and cost.

5.3 Design Choices and Cost

Refinement is useful throughout the trajectory. QR (First) improves all four agents but consistently underperforms QR (All): +2.75 vs. +3.65 points for Search-R1, +0.63 vs. +1.78 for R1-Searcher, +3.38 vs. +6.00 for HiPRAG, and +2.65 vs. +6.18 for ReSearch (Table 1). Query unreliability therefore arises beyond the first search step, motivating confidence checks throughout the trajectory.

Minimum confidence is the strongest tested aggregation. Among trajectory-level aggregations of query-token probabilities, the minimum achieves the highest AUROC for final-answer correctness (0.631), ahead of the geometric mean (0.579), arithmetic mean (0.577), and median (0.562). This supports the weakest-link motivation in Sec. 3.2. Full definitions and ROC analysis are in App. G.5.

Selective refinement adds moderate overhead. QC→QR leaves the number of external search calls nearly unchanged (Table 2) while increasing tool productivity for every agent (App. Tab. 13). Refinement is triggered for 11.2%–47.8% of queries, adding 0.76–2.63 reformulator generations per question (App. Tab. 12). For Search-R1, runtime increases from 17 s to 21.10 s per question, substantially less than Refine All (39.77 s) or Rejection Sampling (33.83 s) (App. Tab. 14).

Additional results, limitations, qualitative trajectories and informal intuition **are in the appendix.**

6 Conclusion

We study *query unreliability* in search-augmented reasoning: an agent may correctly decide to search yet issue a query that poorly expresses its current information need. We show that the agent's own token-level query confidence provides a useful pre-retrieval signal of this failure and introduce *Confidence-Guided Query Refinement (QC→QR)*, a post-hoc test-time method that selectively repairs low-confidence queries while leaving confident ones unchanged.

Across 4 heterogeneous search agents and 4 knowledge-intensive QA benchmarks, QC→QR consistently improves answer accuracy while maintaining a comparable number of external search calls. The gains are accompanied by improved gold-document retrieval, less redundant retrieval, and better downstream calibration, and extend to open-web evaluation. Importantly, unconditional, random, and resampling-based refinement do not improve consistently across agents: the gains come from combining the decision of *when* to intervene with a targeted repair of the query, rather than from refining every query.

Because QC→QR needs only the agent's token probabilities and an off-the-shelf reformulator, it can be added to existing search agents without retraining the agent or the retriever, including agents already trained with process rewards. More broadly, our results suggest that an agent's confidence in an action it is about to take can guide whether to repair that action, not only whether to take it, motivating confidence-guided repair for other tool-augmented reasoning systems in which the agent writes the tool input, such as code and API calls.

AI USE STATEMENT

Generative AI tools were used as supporting research and writing aids in this work. In particular, ChatGPT (GPT-5.6) was used to assist with manuscript drafting and language editing, LaTeX formatting, tables, feedback on experimental design and analysis, and refinement of mathematical arguments. Claude Opus 5.5 (Anthropic), used through Claude Code, assisted with restructuring and editing the manuscript, checking the consistency of reported numbers across tables and text, checking descriptions of related work against the cited papers, and LaTeX formatting. We also used the Stanford Agentic Reviewer (https://paperreview.ai/) and the CMU Paper Reviewer (Kim et al., 2026) as author-side review tools to obtain preliminary critical feedback and to identify aspects of the manuscript requiring further clarification, experimentation, or validation.

All AI-assisted suggestions were critically reviewed by the authors before inclusion. Experimental results were produced and verified by the authors, and all mathematical statements, citations, and technical claims were independently checked by the authors. Generative AI was not used as an author and did not make autonomous decisions regarding the scientific conclusions of the work. The authors take full responsibility for the final content, experimental methodology, analyses, claims, and conclusions of the paper.

ETHICS STATEMENT

This work studies test-time query refinement for search agents on public question-answering benchmarks (NQ, HotpotQA, 2WikiMultiHopQA, MuSiQue, GAIA, and WebWalkerQA), using publicly released model checkpoints and a public Wikipedia corpus; it involves no human subjects and no personal data. Open-web experiments issue queries through a commercial search API, and retrieved web content may contain inaccurate or biased information. Query confidence is an imperfect proxy for reliability, so QC→QR should complement rather than replace verification in high-stakes applications (App. J).

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 (App. A) specify QC→QR, including the confidence measure, the threshold rule, the refinement operations, and candidate selection. Section 4 lists the search agents and their public checkpoints, the retriever and corpus, the reformulation model, generation settings, thresholds, and seeds. Appendix E describes the datasets and evaluation subsets, threshold selection, the test-time controls, and all metrics, including the LLM-judge setup, and Appendix F gives each agent's prompt format. We will release our code, the question IDs of the evaluation subsets, and additional prompts upon acceptance, subject to the necessary institutional approvals.

REFERENCES

Paul J. L. Ammann, Jonas Golde, and Alan Akbik. Question decomposition for retrieval-augmented generation. In Jin Zhao, Mingyang Wang, and Zhu Liu (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 497–507, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-254-1. doi: 10.18653/v1/2025.acl-srw.32. URL https://aclanthology.org/2025.acl-srw.32/.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. Context-DPO: Aligning language models for context-faithfulness. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10280–10300, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.536. URL https://aclanthology.org/2025.findings-acl.536/.

Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. Parameters vs. context: Fine-grained control of knowledge reliance in language models, 2025b. URL https://arxiv.org/abs/2503.15888.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation, 2024. URL https://arxiv.org/abs/2404.00610.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2503.19470.

Yuqin Dai, Shuo Yang, Guoqing Wang, Yong Deng, Zhanwei Zhang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Changhua Meng, Can Yi, Yuchen Zhou, Weiqiang Wang, and Shuai Lu. Careful queries, credible results: Teaching rag models advanced web search tools with reinforcement learning, 2026. URL https://arxiv.org/abs/2508.07956.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, Yang Qin, Yuan Wang, Quanxing Zha, Sunhao Dai, and Changhua Meng. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward, 2025. URL https://arxiv.org/abs/2508.12800.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.16410.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Linfeng Gao, Qinggang Zhang, Baolong Bi, Bo Zeng, Zheng Yuan, Zerui Chen, Zhimin Wei, Shenghua Liu, Linlong Xu, Longyue Wang, Weihua Luo, and Jinsong Su. Beyond black-box interventions: Latent probing for faithful retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2510.12460.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. URL https://arxiv.org/abs/1706.04599.

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

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity, 2024. URL https://arxiv.org/abs/2403.14403.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation, 2023. URL https://arxiv.org/abs/2305.06983.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Seungone Kim, Dongkeun Yoon, Kiril Gashteovski, Juyoung Suk, Jinheon Baek, Pranjal Aggarwal, Ian Wu, Viktor Zaverkin, Spase Petkoski, Daniel R Schrider, et al. On the limits and opportunities of ai reviewers: Reviewing the reviews of nature-family papers with 45 expert scientists. *arXiv preprint arXiv:2605.20668*, 2026.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

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/.

Zheng Liu, Longxiang Zhang, Xintong Wang, Zhiang Xu, Shaoxiong Zhan, Xin Shan, Wen Huang, Tao Dai, Shu-Tao Xia, Chengfu Huo, and Liang Ding. Arbor: Online process rewards via a reusable rubric buffer for search agents, 2026b. URL https://arxiv.org/abs/2606.03239.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. RaFe: Ranking feedback improves query rewriting for RAG. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 884–901, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.49. URL https://aclanthology.org/2024.findings-emnlp.49/.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Bohan Zhang, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz

Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wenting Zhan, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2026. URL https://arxiv.org/abs/2412.16720.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Smart: Self-aware agent for tool overuse mitigation, 2025. URL https://arxiv.org/abs/2502.11435.

Yuanhao Shen, Xiaodan Zhu, and Lei Chen. SMARTCAL: An approach to self-aware tool-use evaluation and calibration. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 774–789, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.59. URL https://aclanthology.org/2024.emnlp-industry.59/.

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

Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher++: Incentivizing the dynamic knowledge acquisition of llms via reinforcement learning, 2025b. URL https://arxiv.org/abs/2505.17005.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.702/.

Nishant Subramani, Jason Eisner, Justin Svegliato, Benjamin Van Durme, Yu Su, and Sam Thomson. Mice for cats: Model-internal confidence estimation for calibrating agents with tools, 2025. URL https://arxiv.org/abs/2504.20168.

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, 2026. URL https://arxiv.org/abs/2604.17337.

Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Song Yang, and Han Li. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding, 2025. URL https://arxiv.org/abs/2501.07861.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze

Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023. URL https://arxiv.org/abs/2212.10509.

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, 2026. URL https://arxiv.org/abs/2510.14967.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Acting less is reasoning more! teaching model to act efficiently, 2025a. URL https://arxiv.org/abs/2504.14870.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization, 2025b. URL https://arxiv.org/abs/2505.15107.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents, 2026. URL https://arxiv.org/abs/2601.04888.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025a. URL https://arxiv.org/abs/2501.07572.

Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Chen. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19723–19734, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.998. URL https://aclanthology.org/2025.emnlp-main.998/.

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*, volume 2026, pp. 82317–82347, 2026.

Roy Xie, Deepak Gopinath, David Qiu, Dong Lin, Haitian Sun, Saloni Potdar, and Bhuwan Dhingra. Over-searching in search-augmented large language models. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7714–7739, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.361. URL https://aclanthology.org/2026.eacl-long.361/.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024. URL https://arxiv.org/abs/2306.13063.

Ruotao Xu, Yixin Ji, Yu Luo, Jinpeng Li, Dong Li, Peifeng Li, Juntao Li, and Min Zhang. When to trust tools? adaptive tool trust calibration for tool-integrated math reasoning, 2026. URL https://arxiv.org/abs/2604.08281.

Boyang Xue, Bin Wu, Shuofei Qiao, Sheng Wang, Rui Wang, Yiming Du, Hongru Wang, Jeff Z. Pan, Emine Yilmaz, Kam-Fai Wong, and Aldo Lipani. Mitigating context interference for reliable and efficient search agents. 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. 3541–3558, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.160. URL https://aclanthology.org/2026.acl-long.160/.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2601.06842.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. FaithfulRAG: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21863–21882, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1062. URL https://aclanthology.org/2025.acl-long.1062/.

APPENDIX CONTENTS

Due to space constraints, several supporting details could not be included in the main paper, including an informal account of why confidence-gated repair works, metric definitions, extended related work, implementation details of benchmark methods, and additional analyses. To keep the submission self-contained and support reproducibility, the supplementary material provides these details together with descriptions of the datasets, hyperparameter settings, baseline configurations, and further experimental results on confidence-guided query reformulation.

Appendix Table of Contents

Section		Page
A	**Algorithm**	**17**
B	**Notation**	**17**
C	**Why Does Confidence-Gated Repair Work?**	**17**
D	**Extended Related Work**	**19**
E	**Experimental Details**	**20**
E.1	Dataset Details	20
E.2	Threshold Selection and Test-Time Controls	22
E.3	Metrics	22
F	**Search Agents and Prompt Formats**	**24**
F.1	Search-R1	24
F.2	R1-Searcher	25
F.3	HiPRAG	26
F.4	ReSearch	26
G	**Additional Results**	**27**
G.1	Sensitivity to the Random Seed	27
G.2	Refinement Frequency and Cost	28
G.3	Calibration	28
G.4	Trajectory Confidence and Accuracy	28
G.5	ROC Analysis of Confidence Aggregation	28
H	**Qualitative Analysis of Query Refinement**	**31**
H.1	Case Study 1: Multi-Hop Genealogical Reasoning	31
H.2	Case Study 2: Resolving Setting–Release-Year Ambiguity	32
H.3	Case Study 3: Tracing a Source Novel Through an Adaptation	33
H.4	Case Study 4: Resolving Series–Spin-off Ambiguity	34
I	**Limitations**	**36**
J	**Broader Impact and Implications**	**36**

A ALGORITHM

Input: question x; search policy π_θ; reformulator $\mathcal{M}_R$; retriever $\mathcal{R}$; threshold λ
Output: final answer A

$s_1 \leftarrow \{x\}$; $t \leftarrow 1$; $\mathcal{G} \leftarrow \{g_{\text{rew}}, g_{\text{dec}}, g_{\text{dis}}\}$
while no final answer has been generated **do**
  $a_t \sim \pi_\theta(\cdot \mid s_t)$ ▷ next action: a query or a final answer
  if a_t is a final answer **then**
    $A \leftarrow a_t$; **break**
  end if
  $q_t \leftarrow a_t$; $C_t \leftarrow \min_i P_\theta(q_{t,i} \mid s_t, q_{t,<i})$ ▷ query confidence, Eq. 4
  if $C_t < \lambda$ **then** ▷ **uncertain**: refine before retrieval
    for $g \in \mathcal{G}$ **do**
      $q_t^{(g)} \sim \mathcal{M}_R(\cdot \mid s_t, q_t, g)$ ▷ rewrite, decompose, disambiguate
    end for
    $\mathcal{Q}_t^R \leftarrow \{q_t^{(g)} : g \in \mathcal{G}\}$
    $q_t^\star \leftarrow \arg\max_{q \in \mathcal{Q}_t^R} C_R(q)$ ▷ most confident candidate, Eq. 8
    $q_t^{\text{exec}} \leftarrow q_t^\star$
  else ▷ **confident**: execute unchanged
    $q_t^{\text{exec}} \leftarrow q_t$
  end if
  $d_t^\star \leftarrow \mathcal{R}(q_t^{\text{exec}})$ ▷ same retriever as the base agent
  $s_{t+1} \leftarrow f_\theta(s_t, q_t^{\text{exec}}, d_t^\star)$; $t \leftarrow t+1$
end while
return A

Algorithm 1: **Confidence-Guided Query Refinement (QC→QR).** At each search step, a query whose confidence falls below λ is refined before retrieval: the reformulator proposes one candidate per operation (rewrite, decompose, disambiguate), and the most confident candidate is executed. Confident queries are executed unchanged, and the executed query is submitted to the same retriever. Restricting the check to $t = 1$ gives the first-query-only variant, QR (First). Table 7 summarizes the notation.

B NOTATION

Table 7 lists the symbols used for the method (Sec. 3 and Algorithm 1), and Table 8 lists those used for final-answer calibration (App. C).

C WHY DOES CONFIDENCE-GATED REPAIR WORK?

This section gives an informal account of why confidence-gated repair improves search agents, and why the alternatives in Tab. 2 do not. Each point is tied to results reported in the paper; the account is an interpretation of those results rather than a formal guarantee.

Refinement is not free, so the targets matter. Refining a query helps when the query misstates the agent's information need, but it can hurt a query that was already right: a rewrite can drift from the agent's intent, drop a constraint, or decompose a question that did not need it. Whether refinement helps therefore depends on which queries receive it. The agent's own query confidence C_t indicates where to look: low-confidence queries retrieve supporting evidence less often (Fig. 3), and refining them improves retrieval (Tab. 3). The controls are consistent with this. Refining every query lowers accuracy for three of the four agents, and refining a random set of queries of the same size lowers it for all four; only aiming the repair at low-confidence queries improves all four (Tab. 2).

Resampling repeats the agent's own misunderstanding. Rejection Sampling uses the same gate, but repairs a flagged query by drawing a new one from the agent itself. A low-confidence query often reflects how the agent has framed the question, for example by committing to the wrong relation or skipping an intermediate entity (App. H). New samples from the same policy tend to share that framing, and a sample that clears the threshold can still be wrong. The reformulator provides a

Table 7: **Notation for the method.** Symbols used in Sec. 3 and Algorithm 1.

Symbol	Meaning
x	Input question or problem instance.
π_θ	Search-agent policy parameterized by θ.
s_t	Agent state at search step t, containing the question and accumulated reasoning and retrieval history.
a_t	Action generated by π_θ at step t, corresponding to a search query or final answer in Algorithm 1.
q_t	Search query generated by the agent at search step t.
$q_{t,i}$, $q_{t,<i}$	i-th token of q_t and its preceding query-token prefix, respectively.
n_t	Number of tokens in query q_t.
$C_t = C(q_t)$	Query confidence at step t, defined as the minimum generated-token probability under π_θ.
λ	Validation-selected confidence threshold controlling whether q_t is executed unchanged or refined.
T	Number of search steps in a reasoning trajectory.
τ	Complete search–reasoning trajectory.
C_τ	Trajectory-level query confidence, $C_\tau = \min_{1\le t\le T} C_t$; used for analysis rather than intervention.
$\mathcal{M}_R$	Reformulation model used to generate candidate query refinements.
$\mathcal{G}$	Set of refinement operations, $\{g_{\mathrm{rew}}, g_{\mathrm{dec}}, g_{\mathrm{dis}}\}$, corresponding to rewrite, decomposition, and disambiguation.
g	A refinement operation in $\mathcal{G}$.
$q_t^{(g)}$	Candidate refinement of q_t generated using operation g.
$\mathcal{Q}_t^R$	Set of candidate refinements generated for query q_t.
$C_R(q)$	Generation confidence assigned by $\mathcal{M}_R$ to candidate refinement q.
$q_t^\star$	Highest-confidence refined query selected from $\mathcal{Q}_t^R$; defined when refinement is triggered.
q_t^{exec}	Query actually submitted to the retriever: q_t if $C_t \ge \lambda$, and $q_t^\star$ otherwise.
$\mathcal{R}$	External retriever or search function.
d_t	Evidence retrieved by the unmodified agent, $d_t = \mathcal{R}(q_t)$.
$d_t^\star$	Evidence retrieved after confidence-guided control, $d_t^\star = \mathcal{R}(q_t^{\mathrm{exec}})$.
f_θ	State-update function that integrates the current state, executed query, and retrieved evidence.

Table 8: **Notation for final-answer calibration.** Symbols used in App. C and Sec. 5.2.

Symbol	Meaning
$Y \in \{0, 1\}$	Indicator of final-answer correctness.
$S \in [0, 1]$	Confidence assigned to the final answer.
ECE, ACE	Expected and adaptive calibration error, computed from final-answer confidence.

different source of queries, and its operations (rewriting, decomposition, and disambiguation) target exactly these failures. Consistent with this, Rejection Sampling lowers accuracy for all four agents, whereas the same gate paired with the reformulator improves all four (Tab. 2).

The warning sign exists only before retrieval. Query confidence C_t and answer confidence S are measured at different moments. C_t is measured before retrieval, while the agent is still deciding what to look for. S is measured after retrieval, conditioned on the passages the agent has read, so it reflects how well the answer is supported by that evidence rather than whether the evidence was right for the question. Once an unreliable query has retrieved plausible but wrong passages, the agent can read an answer off them with high confidence: in Fig. 1, Search-R1 answers "Jasper" because the retrieved passage names Tom the cat "Jasper". The warning sign is visible only while the query is being written, which is why QC→QR checks the query rather than the answer.

Why every query is checked. Unreliable queries are not confined to the first search step. As the agent gathers partial evidence, it forms new information needs, and the queries it writes for them can be just as underspecified as the first. Checking only the first query improves all four agents, but checking every query improves them more (Tab. 1).

Why answer calibration improves. The same mechanism offers an account of the calibration results for Search-R1 (Tab. 6, Fig. 5). For confidence bins b with weights w_b, $\mathrm{ECE} = \sum_b w_b \,|\mathrm{conf}(b) - \mathrm{acc}(b)|$. Since $\sum_b w_b \,\mathrm{conf}(b) = \mathbb{E}[S]$ and $\sum_b w_b \,\mathrm{acc}(b) = \mathbb{E}[Y]$, where $Y \in \{0, 1\}$ denotes answer correctness, the triangle inequality gives

$$\mathrm{ECE} \;\geq\; \big|\mathbb{E}[S] - \mathbb{E}[Y]\big|, \tag{9}$$

with equality if and only if no two nonempty bins err in opposite directions. Search-R1 is close to this regime: accuracy lies below confidence in almost every bin, and ECE and ACE nearly coincide in every setting. Its ECE is therefore close to the overconfidence gap in both runs, so that $\Delta\mathrm{ECE} \approx \Delta\mathbb{E}[S] - \Delta\mathbb{E}[Y]$. Pooled over the four benchmarks, accuracy rises from 0.372 to 0.409 and ECE falls from 0.400 to 0.343, implying that mean confidence falls from about 0.772 to 0.752 (these values are exact when every bin is overconfident, and upper bounds otherwise). Roughly two thirds of the reduction in ECE thus comes from higher accuracy and one third from lower mean confidence. This matches the account above: QC→QR does not change how S is computed, but it changes the evidence on which confident answers are based, so the reduction reflects changes in answer correctness and confidence rather than a recalibration of the confidence estimator itself. Our results are consistent with this mechanism, although they do not isolate it.

D Extended Related Work

Training Search Agents. Search agents learn to interleave reasoning with retrieval through reinforcement learning on final-answer correctness (Jin et al., 2025; Song et al., 2025a; Chen et al., 2025). Search-R1 optimizes an outcome reward while masking retrieved tokens from the policy loss, R1-Searcher first rewards retrieval invocation and then answer correctness, and ReSearch treats search operations as part of the reasoning chain. Because outcome-only rewards can suffer from sparse credit assignment and poor sample efficiency, recent work introduces process-level signals. Information-gain-based optimization (Wang et al., 2026) derives intrinsic rewards from belief updates, StepSearch (Wang et al., 2025b) uses step-wise rewards to encourage informative retrieval and discourage redundancy, and HiPRAG (Wu et al., 2026) uses hierarchical process rewards that assess each search decision. A related line targets unnecessary or redundant tool use (Wu et al., 2025b) through trajectory-level rewards (Wang et al., 2025a), process reward models (Sun et al., 2025), and self-aware tool-use policies (Qian et al., 2025; Shen et al., 2024), among other reinforcement-learning approaches (Sun et al., 2026; Deng et al., 2025; Feng et al., 2025). All of these methods change the agent's parameters to shape *when* and *how often* it searches; the query itself is supervised only through downstream rewards. QC→QR instead operates on an agent that has already been trained, without modifying it, and improves agents trained with either outcome or process rewards, including HiPRAG (Sec. 5.1).

Failure Modes of Search Agents. Despite strong performance, search agents remain vulnerable to failures throughout the search–reasoning pipeline, including poorly or uncertainly formulated queries

(Wu et al., 2025b), interference from irrelevant retrieved context (Xue et al., 2026), over-searching (Xie et al., 2026), and retrieval of evidence that is semantically relevant yet insufficient for the current information need (Liu et al., 2026a). Even when useful evidence is retrieved, models may ignore or inconsistently use it, particularly when retrieved information conflicts with parametric knowledge (Zhang et al., 2025; Gao et al., 2026; Ye et al., 2026; Bi et al., 2025a;b), and they may misjudge when to trust tool outputs (Xu et al., 2026). Most of these failures arise once evidence is already in the agent's context. We target one that arises earlier: an unreliable query can prevent the needed evidence from ever being retrieved, and QC→QR detects it before the query is executed.

Improving Search Queries. Query rewriting is a standard component of RAG systems. RQ-RAG (Chan et al., 2024) trains a model to refine queries by rewriting, decomposing, and disambiguating them, and RaFe (Mao et al., 2024) trains a query rewriter from the feedback of a publicly available reranker instead of annotations. For search agents, SmartSearch (Wen et al., 2026, contemporaneous) assigns process rewards to each intermediate query and refines low-quality queries during training, so that the agent learns to generate better ones, and WebFilter (Dai et al., 2026) uses reinforcement learning to teach agents to issue source-restricted web queries and filter unreliable results. These methods either rewrite every query or train the agent to improve its own queries. QC→QR does neither: it leaves the agent unchanged and decides at test time *which* of its queries to rewrite, using RQ-RAG only as an off-the-shelf reformulator. The distinction matters in practice: applying the same reformulator to every query lowers or leaves unchanged accuracy for three of the four agents we evaluate (Tab. 2).

Uncertainty in Retrieval and Search. Model uncertainty has also been used to control retrieval. FLARE (Jiang et al., 2023) retrieves when a tentative next sentence contains low-probability tokens, and forms the query by masking those tokens or generating questions about them; DRAGIN (Su et al., 2024) combines token uncertainty with attention to decide when to retrieve and what to retrieve. Search Wisely (Wu et al., 2025b) measures an agent's confidence by the minimum token probability over the search queries in a rollout and uses it as a threshold in the training reward. Beyond retrieval, MICE (Subramani et al., 2025) learns a confidence estimator from a model's intermediate layers to decide whether a tool-using agent should make a tool call, while confidence that LLMs state verbally tends to be overconfident (Xiong et al., 2024). QC→QR differs from these methods in what it checks and what it does with the signal. It checks an explicit query issued by a trained search agent, rather than the model's own tentative output; it uses the agent's raw token probabilities at test time, rather than a training reward or a learned estimator; and it repairs a low-confidence query instead of only deciding whether to act. The improved final-answer calibration we report (Tab. 6) also arises without any post-hoc recalibration of output confidence, such as temperature scaling (Guo et al., 2017).

E Experimental Details

E.1 Dataset Details

Datasets and evaluation splits. We evaluate on four open-domain question-answering benchmarks: Natural Questions–Open (NQ-Open) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2023). These datasets cover both single-hop information seeking and multi-hop reasoning over multiple documents; Table 9 summarizes their principal characteristics.

Because every agent must be run as the original policy, with both QC→QR variants, and with all test-time control baselines, we evaluate on fixed random subsets rather than full splits, following common practice in search-augmented QA. For Search-R1, we sample 500 questions per dataset, matching the 500-question evaluation subsets used by IRCoT (Trivedi et al., 2023), Adaptive-RAG (Jeong et al., 2024), and R1-Searcher++ (Song et al., 2025b). For R1-Searcher, HiPRAG, and ReSearch, we sample 200 questions per dataset, consistent with the fixed 200-question multi-hop splits used in recent agentic RL work (Liu et al., 2026b), which follow the evaluation split released with Tool-Star (Dong et al., 2025)[1] and adopted by ARPO (Dong et al., 2026); we subsample NQ-Open at the same size. All subsets are drawn with a fixed seed (42) from the development sets of HotpotQA, 2WikiMultiHopQA, and MuSiQue and from the test set of NQ-Open. While evaluating HiPRAG,

[1] https://github.com/RUC-NLPIR/Tool-Star

ReSearch, and R1-Searcher, some examples encountered retrieval failures; these cases were excluded from the accuracy calculation. HiPRAG on MuSiQue uses 135/200 questions, R1-Searcher on NQ uses 191/200, and ReSearch on HotpotQA uses 197/200 for answer-accuracy evaluation.

Table 9: Representative examples illustrating the reasoning requirements of the evaluated datasets. NQ-Open primarily requires direct single-hop retrieval, whereas HotpotQA, 2WikiMultiHopQA, and MuSiQue require evidence composition across multiple retrieval steps.

Dataset	Example Question	Required Reasoning	Answer
NQ-Open	When is the last episode of Season 8 of *The Walking Dead*?	Retrieve the relevant Wikipedia information and directly identify the broadcast date of the final episode. This is primarily a single-hop lookup.	April 15, 2018
HotpotQA	Which magazine was started first, *Arthur's Magazine* or *First for Women*?	Retrieve the founding year of *Arthur's Magazine*, retrieve the founding year of *First for Women*, and compare the two dates.	Arthur's Magazine
2Wiki	What record label did the person who is part of *The Bruce Lee Band* start?	First identify the person associated with *The Bruce Lee Band* as Mike Park. Then retrieve the record label founded by Mike Park.	Asian Man Records
MuSiQue	Where did the director of *Baldevins bryllup* die?	First identify the director of *Baldevins bryllup* as George Schnéevoigt. Then determine George Schnéevoigt's place of death.	Copenhagen

Natural Questions–Open. Natural Questions (NQ) was constructed from anonymized, aggregated queries submitted by users to the Google search engine. For each query, annotators were shown a Wikipedia page from the top search results and asked to identify a long answer, usually a paragraph, and a short answer, typically one or more entities. The original Natural Questions release contains 307,373 training examples, 7,830 development examples, and 7,842 sequestered test examples (Kwiatkowski et al., 2019).

We use the open-domain variant, NQ-Open, introduced for open-domain question answering. NQ-Open removes the gold Wikipedia document from the model input: given only a question, a system must retrieve relevant evidence and produce an English answer string. All questions are answerable using English Wikipedia.

HotpotQA. HotpotQA is a Wikipedia-based multi-hop QA benchmark containing approximately 113K question–answer pairs (Yang et al., 2018). Each question is designed to require information from multiple documents. The dataset contains two main question categories. *Bridge* questions require identifying an intermediate entity and using it to reach the final answer, whereas *comparison* questions require comparing attributes of two entities. HotpotQA additionally provides sentence-level supporting-fact annotations, enabling evaluation of both final-answer correctness and evidence identification.

The benchmark provides both a *distractor* setting, in which the gold paragraphs are mixed with irrelevant paragraphs, and a *full-wiki* setting, in which systems must retrieve evidence from the complete Wikipedia collection.

2WikiMultiHopQA. 2WikiMultiHopQA was designed to provide explicit supervision for the complete reasoning path of a multi-hop question (Ho et al., 2020). Its construction combines unstructured Wikipedia passages with structured Wikidata relations. Question templates and logical rules are used to ensure that intermediate reasoning steps are necessary rather than optional artifacts of the question-generation process. The complete dataset contains 192,606 examples.

The dataset defines four question types: *inference*, *compositional*, *comparison*, and *bridge-comparison*. In addition to the answer and sentence-level supporting facts, each example can contain an evidence chain represented as subject–relation–object triples.

MuSiQue. MuSiQue was introduced to reduce reasoning shortcuts in existing multi-hop QA benchmarks (Trivedi et al., 2023). It is constructed bottom-up by composing independently answerable single-hop questions. The composition procedure requires the answer of one subquestion to be used in another, producing connected reasoning chains rather than sets of independently answerable facts. MuSiQue-Ans contains approximately 25K answerable questions requiring between two and four reasoning hops.

Table 10: **Accuracy (%) of control methods across agents and datasets.**

Agent	Dataset	Rejection Sampling	Refine All	Random Refinement
ReSearch	NQ	52.0	57.0	51.0
	HotpotQA	53.8	61.9	54.8
	2Wiki	52.0	51.0	48.5
	MuSiQue	36.5	41.0	35.5
R1-Searcher	NQ	50.8	51.8	50.3
	HotpotQA	51.0	56.0	30.0
	2Wiki	52.5	54.0	47.5
	MuSiQue	28.0	30.0	34.0
Search-R1	NQ	25.0	47.0	44.0
	HotpotQA	30.8	35.0	32.6
	2Wiki	27.4	36.0	34.0
	MuSiQue	19.8	23.0	16.6
HiPRAG	NQ	58.0	56.0	57.0
	HotpotQA	51.0	47.0	42.5
	2Wiki	52.5	57.0	59.0
	MuSiQue	28.9	25.9	27.4

Each example includes a multi-hop question, a set of candidate paragraphs, the final answer, paragraph-level supporting annotations, and the constituent single-hop questions used during construction. MuSiQue-Full augments the answerable examples with unanswerable contrast instances.

E.2 Threshold Selection and Test-Time Controls

For threshold selection, we construct a separate validation set of 200 questions per dataset from the development sets, disjoint from all evaluation questions; λ is selected on this set and fixed before test evaluation. We optimized λ on the validation set using final-answer accuracy as the objective. The resulting thresholds are 0.4125, 0.33, 0.36, and 0.175 for Search-R1, R1-Searcher, HiPRAG, and ReSearch, respectively. These thresholds determine which queries are selectively refined: queries with confidence below λ are treated as unreliable and refined, while queries with confidence above the threshold are executed unchanged. Furthermore, as shown by the refine-all baseline in Table 2, indiscriminately refining all queries without accounting for their confidence can degrade final-answer accuracy, highlighting the importance of confidence-guided selective intervention.

We compare QC→QR with three test-time controls that use the same agent, retriever, and questions. *Refine All* applies the reformulator to every search query. *Random Refinement* applies it to randomly chosen queries with the same intervention budget as QC→QR. *Rejection Sampling* keeps the confidence check but replaces the reformulator: whenever $C_t < \lambda$, it resamples the query from the search agent until $C_t \geq \lambda$, for up to five attempts, and keeps the last sample if none passes. Main runs use seed 42; the controls use seed 40 for random selection and generation (Table 10).

E.3 Metrics

Let $\mathcal{D} = \{(x_i, a_i^\star)\}_{i=1}^N$ denote an evaluation set containing N questions, where $a_i^\star$ is the reference answer and $\hat{a}_i$ is the model prediction. Let $y_i \in \{0, 1\}$ denote whether the predicted answer is correct, and let $c_i \in [0, 1]$ denote the confidence assigned to the corresponding prediction.

Answer accuracy. We judge each predicted answer with Qwen3-8B, prompted to decide whether it is semantically consistent with the reference answer, following Chen et al. (2025); $y_i = 1$ if the judge accepts the answer.

Judge Prompt Given a Question and its Golden Answer, verify whether the Predicted Answer is correct. The prediction is correct if it fully aligns with the meaning and key information of the Golden Answer. Respond with True if the prediction is correct and False otherwise. Question: Golden Answer: Predicted Answer:

Tool Productivity. Tool Productivity (TP) measures how effectively external-tool calls are converted into correct final answers. Let T_i denote the number of tool calls made for example i. We define

$$\mathrm{TP} = \frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} T_i}. \tag{10}$$

A higher TP indicates that the agent obtains more correct answers per tool invocation, reflecting more efficient use of retrieval resources.

E.3.1 Calibration Metrics

We also evaluate whether model confidence reflects empirical correctness. For each example i, let $c_i \in [0, 1]$ denote the model's confidence in the predicted answer and $y_i \in \{0, 1\}$ its correctness indicator. We compute c_i as the geometric mean of the probabilities assigned to the L_i tokens of the predicted answer $\hat{a}_i$,

$$c_i = \left(\prod_{j=1}^{L_i} P_\theta \left(\hat{a}_{i,j} \mid s_A, \hat{a}_{i,<j} \right) \right)^{1/L_i}, \tag{11}$$

where s_A is the agent state at which the answer is generated.

Expected Calibration Error (ECE). We partition predictions into M confidence bins $\{B_m\}_{m=1}^{M}$. For each bin, empirical accuracy and average confidence are

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} y_i, \tag{12}$$

and

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} c_i. \tag{13}$$

ECE is the weighted average calibration gap:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right| \tag{14}$$

Lower ECE indicates better calibration. We use fixed-width confidence bins for ECE.

Adaptive Calibration Error (ACE). ACE follows the same principle as ECE, but constructs bins adaptively such that each bin contains approximately the same number of examples. Let $\{\widetilde{B}_m\}_{m=1}^{M}$ denote the resulting equal-frequency bins. Then

$$\mathrm{ACE} = \frac{1}{M} \sum_{m=1}^{M} \left| \mathrm{acc}(\widetilde{B}_m) - \mathrm{conf}(\widetilde{B}_m) \right| \tag{15}$$

Adaptive binning reduces sensitivity to sparsely populated confidence regions.

Maximum Calibration Error (MCE). MCE measures the largest calibration discrepancy among confidence bins:

$$\mathrm{MCE} = \max_{m \in \{1, \ldots, M\}} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right| \tag{16}$$

Unlike ECE, which summarizes average miscalibration, MCE captures the worst calibrated confidence region.

Brier Score. The Brier score measures the mean squared error between confidence and the binary correctness outcome:

$$\mathrm{Brier} = \frac{1}{N} \sum_{i=1}^{N} \left(c_i - y_i \right)^2. \tag{17}$$

The Brier score jointly reflects calibration and predictive discrimination. Lower values indicate better probabilistic predictions.

Negative Log-Likelihood (NLL). For binary correctness, NLL evaluates the log probability assigned to the observed outcome:

$$\mathrm{NLL} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log c_i + (1-y_i)\log(1-c_i)\right]. \tag{18}$$

For numerical stability, confidence values are clipped to $c_i \in [\epsilon, 1-\epsilon]$ for a small constant $\epsilon > 0$. NLL strongly penalizes incorrect predictions made with high confidence; lower NLL indicates better calibrated confidence estimates.

E.3.2 Retrieval Metrics

Hit@k. We report Hit@k at two levels. Per query, it is the fraction of issued queries whose top-k passages contain a gold supporting passage. Per question, where it is computed once for each question. We evaluate (see Tab. 3) whether retrieved passages contain gold supporting evidence Ammann et al. (2025). For example i, let $\mathcal{R}_i^k$ denote the set of titles of the top-k retrieved passages, and let $\mathcal{G}_i$ denote the set of gold supporting-passage titles. **Hit@1** and **Hit@3** measure whether at least one gold supporting passage appears among the top-1 and top-3 retrieved passages, respectively:

$$\mathrm{Hit@}k = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\mathcal{R}_i^k \cap \mathcal{G}_i \neq \emptyset\right], \qquad k \in \{1,3\}. \tag{19}$$

Novel retrieval rate. We measure whether successive searches contribute new evidence rather than repeatedly retrieving previously observed documents. For the retrieval results at search step t, let O_t denote the number of retrieved documents that also appeared in any earlier search step within the same reasoning trajectory. A retrieval step is considered *novel* when $O_t \leq K$, where K denotes the maximum permitted document overlap. We define the Novel Retrieval Rate as

$$\mathrm{NovelRate@}\,K = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}[O_t \leq K], \tag{20}$$

where T is the total number of retrieval steps and $\mathbb{I}[\cdot]$ is the indicator function. We report results for $K \in \{1,2\}$. *Before* refers to the original search-agent trajectories, whereas *After* refers to the trajectories obtained after confidence-guided query refinement. A higher Novel Retrieval Rate indicates lower retrieval redundancy and greater exploration of previously unseen evidence; novelty alone, however, does not guarantee that the retrieved evidence is relevant, so we report it together with Hit@k.

F Search Agents and Prompt Formats

F.1 Search-R1

Search-R1 Jin et al. (2025) is an RL-based search agent that learns to interleave reasoning with external retrieval. The search engine is treated as part of the environment, allowing the policy to issue multiple search queries during a reasoning trajectory. Search-R1 supports both PPO and GRPO training and masks retrieved tokens from the policy loss, since these tokens are produced by the environment rather than by the policy. Training is guided using a simple outcome-based reward based on final-answer correctness.

Prompt format. Search-R1 instructs the model to reason inside `<think>` tags, invoke retrieval by placing a query inside `<search>` tags, and return the final answer inside `<answer>` tags. Retrieved evidence is inserted by the environment inside `<information>` tags. Schematically,

```
Question: {question}

<think>
  reasoning about the current information need
</think>

<search>
  search query
</search>

<information>
  retrieved documents
</information>

<think>
  reasoning using the retrieved evidence
</think>

<answer>
  final answer
</answer>
```

The search–reasoning cycle can be repeated multiple times before the final answer is produced.

F.2 R1-SEARCHER

R1-Searcher employs a two-stage outcome-based RL framework to teach an LLM how to autonomously use an external retrieval system. In the first stage, the model is rewarded for successfully invoking retrieval and following the required output format. In the second stage, the retrieval incentive is removed and the model is optimized using answer correctness together with a format constraint. The authors use a modified Reinforce++ framework and mask retrieved documents when computing the policy loss.

Prompt format. R1-Searcher separates model-generated search queries from environment-provided documents using dedicated tags. Its interaction structure can be summarized as

```
Question: {question}

<think>
  reasoning
  <begin_of_query>
    search query
  <end_of_query>

  <begin_of_documents>
    retrieved documents
  <end_of_documents>

  continued reasoning
</think>

<answer>
  short final answer
</answer>
```

Generation pauses when `<end_of_query>` is produced; the query is sent to the retriever, and the resulting documents are appended before reasoning continues.

F.3 HiPRAG

HiPRAG Wu et al. (2026) extends RL-based agentic retrieval by introducing fine-grained process supervision over individual search decisions. Rather than optimizing only the final outcome, HiPRAG explicitly identifies over-search and under-search behavior and incorporates the quality of these intermediate decisions into a hierarchical process reward. To enable deterministic step-level evaluation, the reasoning trajectory is decomposed into explicitly structured reasoning steps.

Prompt format. Unlike Search-R1, HiPRAG places the complete reasoning trajectory inside a single `<think>` block and divides it into explicit `<step>` blocks. A search step follows the structure

```
<think>
  <step>
    <reasoning>
      reasoning and identification of an information need
    </reasoning>

    <search>
      search query
    </search>

    <context>
      retrieved evidence
    </context>

    <conclusion>
      conclusion obtained in this step
    </conclusion>
  </step>

  <step>
    <reasoning>
      internal reasoning without retrieval
    </reasoning>

    <conclusion>
      conclusion of the step
    </conclusion>
  </step>
</think>

<answer>
  final answer
</answer>
```

This structured representation allows HiPRAG to distinguish search and non-search steps and evaluate whether each retrieval decision was necessary.

F.4 ReSearch

ReSearch Chen et al. (2025) trains LLMs to learn when and how to search using GRPO without requiring supervised reasoning trajectories. During a rollout, model-generated reasoning is interleaved with search queries and retrieved evidence. Retrieved tokens are masked from the RL loss, while the generated reasoning and search-query tokens are optimized. The reward combines final-answer F1 with a format-based signal.

Prompt format. ReSearch uses separate templates for base and instruction-tuned models, but both enforce the same search interaction structure:

Table 11: Accuracy (%) of QR (First) and QR (All) under seeds 42 (main results) and 36.

		First		All	
Agent	**Dataset**	**42**	**36**	**42**	**36**
Search-R1	NQ	52.8	51.4	53.2	53.6
	HotpotQA	42.6	43.0	43.6	44.2
	2Wiki	37.6	38.4	38.8	39.0
	MuSiQue	26.8	26.0	27.8	29.2
	Avg.	39.95	39.70	40.85	41.50
HiPRAG	NQ	63.0	62.5	65.0	66.0
	HotpotQA	51.5	51.0	55.0	56.0
	2Wiki	64.5	63.5	66.5	65.5
	MuSiQue	28.9	28.1	31.9	32.6
	Avg.	51.98	51.28	54.60	55.03

```
<think>
  reasoning about the question
</think>

<search>
  search query
</search>

<result>
  retrieved documents
</result>

<think>
  reasoning using the search result
</think>

<answer>
  The final answer is \boxed{...}
</answer>
```

Whenever the model generates `</search>`, generation is paused, the query is executed against the retrieval environment, and the retrieved evidence is inserted inside `<result>` tags before generation resumes.

G Additional Results

This section reports how often refinement is triggered and what it costs (App. G.2), final-answer calibration (App. G.3), the relation between trajectory confidence and accuracy (App. G.4), and the choice of confidence aggregation (App. G.5).

G.1 Sensitivity to the Random Seed

To check that our results are not specific to one sampling seed, we rerun QR (First) and QR (All) for Search-R1 and HiPRAG with seed 36 (Table 11). Average accuracy changes by at most 0.7 points between seeds, well below the gains of QC→QR over the original agents, and checking every query outperforms checking only the first on every dataset under both seeds.

Table 12: **Query-refinement frequency and inference cost.** *Queries Refined* denotes the percentage of search queries that fall below λ and are refined, while *Refinement Gen./Q* measures the number of additional query-reformulation generations per question.

Agent	Queries Refined (%)	Refinement Gen./Q ↓
Search-R1	31.30	1.475
R1-Searcher	11.23	0.756
HiPRAG	47.78	2.633
ReSearch	25.37	2.481

Table 13: **Retrieval cost, accuracy gain, and tool productivity.**

Agent	Search/Q Orig. → QR	Δ Acc. (pp) ↑	Tool Productivity (%) Orig. → QR
Search-R1	1.57 → 1.61	**+3.65**	23.69 → 25.37
R1-Searcher	2.25 → 2.23	**+1.78**	21.95 → 22.94
HiPRAG	1.84 → 1.94	**+6.00**	26.41 → 28.14
ReSearch	3.26 → 3.23	**+6.18**	15.75 → 17.81

G.2 Refinement Frequency and Cost

Table 12 reports how often each agent's queries fall below the confidence threshold λ and therefore trigger refinement, along with the additional inference cost this incurs. Table 13 reports the change in the number of search calls per question, the corresponding change in answer accuracy, and the resulting change in tool productivity. Table 14 reports the runtime per question for Search-R1 on HotpotQA.

G.3 Calibration

Figure 5 presents reliability diagrams for Search-R1 across all four benchmarks, comparing the original agent (top row) against the same agent after all-query confidence-guided refinement (bottom row).

G.4 Trajectory Confidence and Accuracy

Figure 6 plots final-answer accuracy against trajectory confidence $C_\tau = \min_t C_t$ for R1-Searcher and ReSearch. Accuracy generally increases with C_τ, consistent with Wu et al. (2025b). This analysis is at the trajectory level; QC→QR itself acts on the per-query confidence C_t.

G.5 ROC Analysis of Confidence Aggregation

We examine how query-token probabilities should be aggregated when summarizing confidence over a search trajectory. Let $\{p_1, \ldots, p_N\}$ denote the probabilities assigned to all N generated query tokens in a trajectory. We compare four aggregation rules.

The **minimum confidence** is defined as

$$C_{\min} = \min_{i=1,\ldots,N} p_i,$$

which captures the least confident query token generated anywhere in the trajectory.

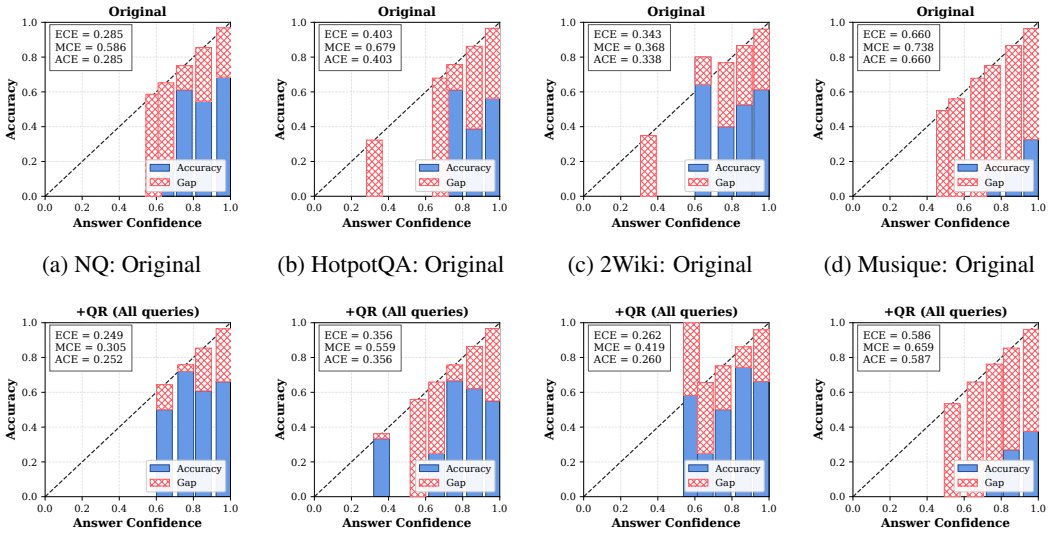
The **arithmetic-mean confidence** is defined as

$$C_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} p_i,$$

which measures the average token-level confidence across the complete search trajectory.

Table 14: Estimated runtime per question for Search-R1 on HotpotQA.

Method	Time per question (s)
Original	17
Rejection sampling	33.83
Refine all	39.77
QR (ours)	21.10



(a) NQ: Original (b) HotpotQA: Original (c) 2Wiki: Original (d) Musique: Original

(e) NQ: +QR (f) HotpotQA: +QR (g) 2Wiki: +QR (h) Musique: +QR

Figure 5: **Confidence-guided query refinement improves downstream answer calibration.** Reliability diagrams for Search-R1 compare final-answer confidence before (top row) and after all-query confidence-guided query refinement, **QR** (bottom row), on NQ, HotpotQA, 2Wiki, and MuSiQue benchmarks. Although **QR** operates only on the confidence of intermediate search queries, the resulting trajectories exhibit smaller confidence–accuracy gaps and consistently lower ECE, indicating improved calibration of the final answer, consistent with the analysis in App. C. Blue bars denote empirical accuracy within confidence bins, hatched regions indicate the confidence–accuracy gap, and the dashed diagonal represents perfect calibration. ECE, MCE, and ACE Nixon et al. (2019) are reported for each setting.

The **geometric-mean confidence** is defined as

$$C_{\text{geo}} = \left(\prod_{i=1}^{N} p_i\right)^{1/N} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log p_i\right),$$

which aggregates token probabilities multiplicatively and therefore places greater emphasis on low-probability tokens than the arithmetic mean.

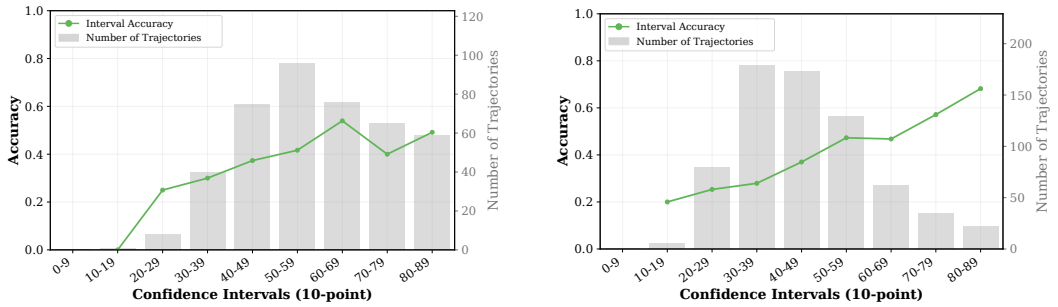
The **median confidence** is defined by first sorting the token probabilities as

$$p_{(1)} \le p_{(2)} \le \cdots \le p_{(N)},$$

and computing

$$C_{\text{median}} = \begin{cases} p_{\left(\frac{N+1}{2}\right)}, & N \text{ odd}, \\ \dfrac{p_{\left(\frac{N}{2}\right)} + p_{\left(\frac{N}{2}+1\right)}}{2}, & N \text{ even}. \end{cases}$$

The minimum is equivalent to taking the minimum query confidence across search steps, $C_\tau = \min_t C_t$; the other aggregations summarize the overall level of confidence rather than the least-confident generation decision.



(a) R1-Searcher (b) ReSearch

Figure 6: **Trajectory confidence and final-answer accuracy.** Trajectory confidence is the minimum query-token probability across all search queries and is grouped into 10-point bins. Green curves show bin accuracy; gray bars show trajectory counts. Higher confidence generally corresponds to higher final-answer accuracy.

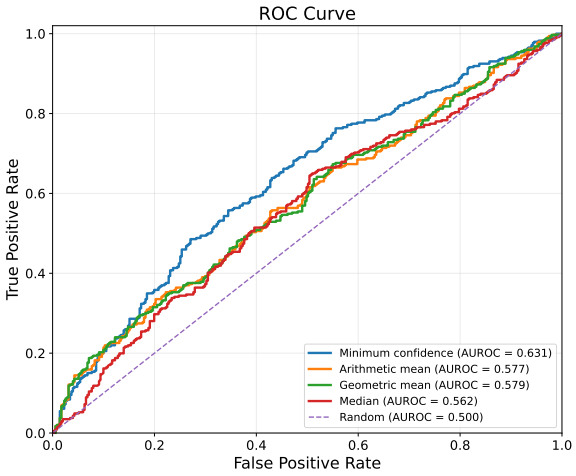


Figure 7: **Minimum confidence best predicts answer correctness.** ROC curves compare alternative aggregations of query-token probabilities over a search trajectory over all datasets for SearchR1 agent. Minimum confidence achieves the highest AUROC (0.631), supporting the weakest-link aggregation used in our method.

Figure 7 evaluates each aggregation as a score for predicting final-answer correctness. Minimum confidence performs best, with an AUROC of **0.631**, compared with **0.579** for the geometric mean, **0.577** for the arithmetic mean, and **0.562** for the median; random discrimination yields an AUROC of 0.5. Thus, averaging confidence across a trajectory weakens the association with answer correctness, whereas retaining the least-confident token provides the most informative signal among the tested aggregations.

This result is consistent with the *weakest-link* interpretation in Sec. 3.2: a trajectory may contain many confidently generated query tokens while a single uncertain entity, relation, or constraint is sufficient to misdirect retrieval and affect the final answer. At the same time, the AUROC of 0.631 indicates that minimum confidence is an informative but imperfect predictor of correctness. We therefore use this trajectory-level analysis only to study the relationship between search uncertainty and downstream outcomes. QC→QR itself operates locally at inference time, using the per-query score C_t to decide whether each query should be executed or refined before retrieval.

H Qualitative Analysis of Query Refinement

H.1 Case Study 1: Multi-Hop Genealogical Reasoning

Question: *Who was the paternal grandfather of Arthur Gore, 2nd Earl of Arran?*

Before QR

Reasoning. I don't have direct access to historical records or databases, so I need to investigate Arthur Gore, 2nd Earl of Arran's paternal grandfather through research. Let's start with a search.

Search: `Who was the paternal grandfather of Arthur Gore, 2nd Earl of Arran?`

Retrieved evidence.

- **Arthur Gore, 2nd Earl of Arran:** Arthur Saunders Gore, 2nd Earl of Arran, was an Irish peer and politician. Arran was the eldest son of Arthur Gore, 1st Earl of Arran, and Jane Saunders.
- **Arthur Gore, 6th Earl of Arran:** Arthur Gore, 6th Earl of Arran, was the only son of Arthur Saunders Gore, 5th Earl of Arran, and Lady Edith Elizabeth Henrietta Jocelyn.
- **Arthur Gore, 1st Earl of Arran:** Arthur Gore, 1st Earl of Arran, was an Irish politician. Arran was the son of Sir Arthur Gore, 2nd Baronet, and Elizabeth Annesley.

Reasoning. The search results provided details about Arthur Gore, 2nd Earl of Arran, but not specifically about his paternal grandfather. However, the information suggests that Arthur Gore, 1st Earl of Arran, was the father of Arthur Gore, 2nd Earl of Arran. Therefore, Arthur Gore, 1st Earl of Arran, would be the paternal grandfather of Arthur Gore, 2nd Earl of Arran.

Final answer: **Arthur Gore, 1st Earl of Arran** ×

After QR

Reasoning. I don't have direct access to historical records or databases, so I need to investigate Arthur Gore, 2nd Earl of Arran's paternal grandfather through research. Let's start with a search.

Search: `Who was Arthur Gore, 2nd Earl of Arran's paternal grandfather?`

Retrieved evidence.

- **Arthur Gore, 2nd Earl of Arran:** Arthur Saunders Gore, 2nd Earl of Arran, was an Irish peer and politician. Arran was the eldest son of Arthur Gore, 1st Earl of Arran, and Jane Saunders.
- **Arthur Gore, 6th Earl of Arran:** Arthur Gore, 6th Earl of Arran, was the only son of Arthur Saunders Gore, 5th Earl of Arran, and Lady Edith Elizabeth Henrietta Jocelyn.
- **Arthur Gore, 1st Earl of Arran:** Arthur Gore, 1st Earl of Arran, was an Irish politician. Arran was the son of Sir Arthur Gore, 2nd Baronet, and Elizabeth Annesley.

Reasoning. The information provided indicates that Arthur Gore, 2nd Earl of Arran's father was Arthur Gore, 1st Earl of Arran. However, the agent does not yet resolve the requested paternal-grandfather relation and therefore performs an additional search.

Search: `Who was Arthur Gore, 1st Earl of Arran's father?` after QR

Retrieved evidence.

- **Arthur Gore, 1st Earl of Arran:** Arthur Gore, 1st Earl of Arran, was an Irish politician. Arran was the son of Sir Arthur Gore, 2nd Baronet, and Elizabeth Annesley.

- **Arthur Gore, 2nd Earl of Arran:** Arthur Saunders Gore, 2nd Earl of Arran, was the eldest son of Arthur Gore, 1st Earl of Arran, and Jane Saunders.
- **Arthur Gore, 1st Earl of Arran:** Arthur Gore, 1st Earl of Arran, was the son of Sir Arthur Gore, 2nd Baronet, and Elizabeth Annesley.

Reasoning. Based on the information gathered, Arthur Gore, 1st Earl of Arran was the father of Arthur Gore, 2nd Earl of Arran. Arthur Gore, 1st Earl of Arran was the son of Sir Arthur Gore, 2nd Baronet. Therefore, Arthur Gore, 2nd Earl of Arran's paternal grandfather was Sir Arthur Gore, 2nd Baronet.

Final answer: Sir Arthur Gore, 2nd Baronet ✓

Effect of QR. Before QR, the retrieved evidence already contains both relations required to answer the question, but the agent fails to compose them correctly and mistakes the father for the paternal grandfather. Following QR, the trajectory introduces a targeted follow-up query that makes the second relational hop explicit:

$$\text{Arthur Gore, 2nd Earl} \rightarrow \text{Arthur Gore, 1st Earl} \rightarrow \text{Sir Arthur Gore, 2nd Baronet.}$$

This explicit decomposition enables the agent to correctly compose the two relations and recover the answer.

H.2 Case Study 2: Resolving Setting–Release-Year Ambiguity

Question: *What year does The Quiet Man take place?*

Before QR

Reasoning. I don't have any initial information about the movie *The Quiet Man* or its release year. I need to find out more details about this film.

Search: `when does the quiet man take place`

Retrieved evidence.

- *The Quiet Man*: In the **1920s**, Sean Thornton travels to Ireland to reclaim his family's farm.
- Filming commenced on June 7, 1951; the story is set in the fictitious community of Inisfree.
- A retrieved result for *The Quiet American* mentions 1952.

Reasoning. The information indicates that the movie is set in the 1920s. However, it does not specify an exact year, so the agent searches again.

Search: `what year does the quiet man take place`

Retrieved evidence.

- The story takes place in the **1920s**.
- Filming commenced in 1951.
- *The Quiet Man* is a **1952** American film.

Reasoning. Although the retrieved evidence states that the film takes place in the 1920s, the model associates the more specific year 1952 with the requested answer.

Final answer: 1952 ×

After QR

Reasoning. I don't have any initial information about the movie *The Quiet Man* or its release year. I need to find out more details about this film.

Search: `When does the film 'The Quiet Man' take place?` QR-refined

Retrieved evidence.

- *The Quiet Man*: In the **1920s**, Sean Thornton travels to Ireland.
- Filming commenced in 1951.
- The unrelated *Quiet American* result mentions 1952.

Reasoning. The retrieved evidence indicates that the film takes place in the 1920s, but the agent seeks confirmation of the temporal setting.

Search: `What year does the film 'The Quiet Man' take place?` QR-refined

Retrieved evidence.

- *The Quiet Man* is a 1952 film.
- The narrative is explicitly set in the **1920s**.
- The film was released in 1952.

Reasoning. The evidence distinguishes the film's release year from its narrative setting. The story takes place in the 1920s.

Final answer: 1920s ✓

Effect of QR. Before QR, the bare keyword query retrieves a distractor document about *The Quiet American* that mentions 1952 alongside the correct 1920s setting, and the agent conflates the film's release year with its narrative setting. Following QR, the disambiguating refinement makes the intended question dimension explicit—narrative setting rather than release year suppressing the distractor and allowing the agent to distinguish the two and recover the correct answer.

H.3 Case Study 3: Tracing a Source Novel Through an Adaptation

Question: *In what year was the novel that Lourenço Mutarelli based Nina on first published?*

Before QR

Reasoning. Identify the novel Lourenço Mutarelli based "Nina" on and its publication year.

Search: `What novel did Lourenço Mutarelli base ''Nina'' on and when was it published?`

Retrieved evidence.

- **Lourenço Mutarelli:** Lourenço Mutarelli is a Brazilian comic artist and writer. In 2004, he wrote the screenplay for the motion picture "Nina," directed by Heitor Dhalia and based on the novel *Crime and Punishment* by Fyodor Dostoyevsky.
- **Lourenço Mutarelli:** In 2004, Mutarelli wrote the screenplay for the motion picture "Nina." The film was based on Fyodor Dostoyevsky's novel *Crime and Punishment*.
- **Nina (2004 film):** "Nina" is a 2004 Brazilian film directed by Heitor Dhalia. The film is loosely based on the novel *Crime and Punishment* by Fyodor Dostoyevsky.

Reasoning. The novel "Nina" by Lourenço Mutarelli is based on *Crime and Punishment* by Dostoyevsky, which was first published in 1866.

Search: `When was the novel Nina by Lourenço Mutarelli published?`

Retrieved evidence.

- **Lourenço Mutarelli:** In 2004, Mutarelli wrote the screenplay for the motion picture "Nina," directed by Heitor Dhalia and based on the novel *Crime and Punishment*.
- **Nina (2004 film):** "Nina" is a 2004 Brazilian film directed by Heitor Dhalia. The film is loosely based on Fyodor Dostoyevsky's novel *Crime and Punishment*.
- **Nina:** "Nina" may refer to several films, books, people, and other works, including the 2004 Brazilian film.

Reasoning. The novel "Nina" by Lourenço Mutarelli was published in 2004.

Final answer: 2004 ×

After QR

Reasoning. Identify the novel Lourenço Mutarelli based "Nina" on.

Search: `What novel did Lourenço Mutarelli base 'Nina' on?` QR-refined

Retrieved evidence.

- **Lourenço Mutarelli:** Lourenço Mutarelli is a Brazilian comic artist and writer. In 2004, he wrote the screenplay for the motion picture "Nina," directed by Heitor Dhalia and based on the novel *Crime and Punishment* by Fyodor Dostoyevsky.
- **Lourenço Mutarelli:** In 2004, Mutarelli wrote the screenplay for the motion picture "Nina." The film was based on Fyodor Dostoyevsky's novel *Crime and Punishment*.
- **Nina (2004 film):** "Nina" is a 2004 Brazilian film directed by Heitor Dhalia. The film is loosely based on the novel *Crime and Punishment* by Fyodor Dostoyevsky.

Reasoning. "Nina" was based on Fyodor Dostoyevsky's novel *Crime and Punishment*.

Search: `When was Crime and Punishment by Dostoyevsky published?` after QR

Retrieved evidence.

- **Crime and Punishment:** *Crime and Punishment* is a novel by the Russian author Fyodor Dostoyevsky. It was first published in the literary journal *The Russian Messenger* in twelve monthly installments during 1866.
- **Crime and Punishment:** The novel was first published in the literary journal *The Russian Messenger* in twelve monthly installments during 1866. It was later published in a single volume.
- **Fyodor Dostoyevsky bibliography:** Dostoyevsky's major works include *Crime and Punishment* (1866), *The Idiot* (1869), *Demons* (1872), and *The Brothers Karamazov* (1880).

Reasoning. *Crime and Punishment* was first published in 1866.

Final answer: 1866 ✓

Effect of QR. Before QR, the agent correctly identifies *Crime and Punishment* as the source novel but then drifts: its second search targets *Nina*'s own publication year rather than that of the source novel, returning 2004. Following QR, the second query is explicitly directed at *Crime and Punishment*, retrieving the correct first-publication year of 1866.

H.4 Case Study 4: Resolving Series–Spin-off Ambiguity

Question: *Which of Tara Strong's major voice roles is an American animated television series based on the DC Comics superhero team "Teen Titans"?*

Before QR

Reasoning. I need to find which of Tara Strong's major voice roles is an American animated television series based on the DC Comics fictional superhero team, the "Teen Titans."

Search: `Tara Strong major voice roles in DC Comics superhero team animated series`

Retrieved evidence.

- **Tara Strong:** Tara Strong is a Canadian-American actress who has done voice work for numerous animations.
- **DC Universe Animated Original Movies:** The DC Universe Animated Original Movies are a series of direct-to-video films based on DC Comics properties.
- **Teen Titans (comics):** The Teen Titans are a fictional superhero team appearing in American comic books published by DC Comics.

Reasoning. The results identify the comic-book team but not which of her roles corresponds to a television series. Let me narrow the search.

Search: `Tara Strong Teen Titans superhero team animated television series role`

Retrieved evidence.

- **Teen Titans (TV series):** Teen Titans is an American animated superhero television series based on the DC Comics characters of the same name.
- **Teen Titans (comics):** The Teen Titans are a fictional superhero team appearing in American comic books published by DC Comics.
- **Tara Strong:** Tara Strong has provided voice work for numerous animated series.

Reasoning. "Teen Titans" is described as based on the DC Comics characters, so that must be the answer.

Final answer: Teen Titans ×

After QR

Reasoning. I need to find which of Tara Strong's major voice roles is an American animated television series based on the DC Comics fictional superhero team, the "Teen Titans." I will first identify her major voice roles, then determine which matches.

Search: `What are Tara Strong's major voice roles?` QR-refined

Retrieved evidence.

- **Tara Strong:** Tara Strong is a Canadian-American actress who has done voice work for numerous animations. Her major voice roles include Rugrats, The Powerpuff Girls, The Fairly OddParents, Ben 10, Teen Titans, and its spin-off Teen Titans Go!
- **Tara Strong filmography:** Tara Strong's major voice roles include Teen Titans and the spin-off series Teen Titans Go!
- **Tara Strong:** Her major voice roles include Teen Titans and its spin-off Teen Titans Go!

Reasoning. Tara Strong voiced roles in both "Teen Titans" and "Teen Titans Go!" I need to determine which one is described as being based on the DC Comics fictional superhero *team*, not merely its characters.

Search: `Which animated series is based on the DC Comics fictional superhero team, the Teen Titans?` after QR

Retrieved evidence.

- **Teen Titans Go! (TV series):** Teen Titans Go! is an American animated television series airing on Cartoon Network since April 23, 2013 and based on the DC Comics fictional superhero team. It is a comedic spin-off of the previous series.
- **Teen Titans (TV series):** Teen Titans is an American animated superhero television series based on the DC Comics *characters* of the same name.
- **Teen Titans (season 1):** Teen Titans is based on the DC Comics series and stars Tara Strong as one of the main characters.

Reasoning. The evidence draws a precise distinction: "Teen Titans Go!" is described as based on the DC Comics fictional *superhero team*, which matches the question's phrasing exactly, whereas "Teen Titans" is described as based on the DC Comics *characters*. Therefore the answer is Teen Titans Go!

Final answer: Teen Titans Go! ✓

Effect of QR. Before QR, the over-specified compound query never retrieves the spin-off; the agent commits to the first partially-matching title ("Teen Titans") without discovering that a second candidate exists. Following QR, the trajectory decomposes into two targeted steps: the first query enumerates all of Tara Strong's major voice roles, surfacing both "Teen Titans" and "Teen Titans Go!" as candidates; the second query resolves the series–spin-off ambiguity by matching the question's precise phrasing (*superhero team* vs. *characters*), recovering the correct answer.

I LIMITATIONS

Several aspects of our setup define the scope of these findings. QC→QR relies on the search policy's token probabilities, so it applies to agents whose output probabilities are accessible, and its threshold λ is selected per agent on validation data. Query confidence is an informative but imperfect signal of unreliable queries (App. G.5): a confidently written query can still be wrong, and QC→QR repairs only the queries it flags. The reformulator is a separately trained model, and refinement adds inference cost, from 17 to 21.1 seconds per question for Search-R1 on HotpotQA (Tab. 14). Our evaluation covers four agents on Wikipedia-based QA and two open-web benchmarks; the calibration and retrieval-novelty analyses cover Search-R1, and each configuration is run once on fixed evaluation subsets. Extending the gate to other tools, and learning when to intervene rather than thresholding, are natural next steps.

J BROADER IMPACT AND IMPLICATIONS

At a narrow level, our results suggest that query confidence can serve as a lightweight reliability signal for improving search-augmented reasoning: uncertain search actions can be selectively reconsidered before retrieval, improving answer quality and downstream calibration without retraining the underlying agent. More broadly, this work motivates a reliability-aware view of agentic reasoning in which models assess not only *which action to take*, but also *whether that action should be trusted before execution*. Although we study search queries, the same principle may extend to other tool-using agents that interact with APIs, databases, code interpreters, mathematical tools, or external services. Such metacognitive control could reduce the propagation of unreliable intermediate actions in high-stakes workflows. At the same time, confidence is an imperfect proxy for correctness, and selective intervention should complement rather than replace external verification, especially in safety-critical applications.