RE-GUIDE: Guiding Retrieval and Reasoning for Efficient Agentic RAG with Reasoning Models

Anonymous ACL submission

Abstract

Agentic RAG extends large reasoning models to knowledge-intensive tasks by interleaving reasoning with retrieval, but it often incurs substantial overhead through redundant searches, unproductive exploration, and continued reasoning after sufficient evidence has been obtained. We propose RE-GUIDE, an inference-time guidance framework for Agentic RAG. RE-GUIDE uses an offline-trained retrieval planner to decompose implicit information needs into concrete retrieval targets before search, and a post-retrieval reasoning guide that uses evidence-sufficiency feedback to decide whether the agent should answer, derive an intermediate conclusion, or continue exploration under a controlled budget. Across six open-domain QA evaluations and three open-weight reasoning models, RE-GUIDE on average achieves a relative 3.4% improvement in F1 while reducing agent-generated tokens by 27.2%, achieving a 35.4% improvement in F1 per agent-generated token. Further analyses show that RE-GUIDE performs state-sensitive regulation: it mitigates under-thinking by encouraging additional exploration for under-developed trajectories and over-thinking by curtailing over-extended, ruminative reasoning.

1 Introduction

Large reasoning models (LRMs) have shown strong performance on mathematics, multi-step reasoning, and coding by generating extended reasoning trajectories (Guo et al., 2025; Moshkov et al., 2025; OpenAI, 2024a). These trajectories exhibit System 2-like behaviors such as problem decomposition, reassessment, and exploration of alternative reasoning paths (Guo et al., 2025; Kahneman, 2011; OpenAI, 2024b; Zhang et al., 2026). Recent work extends these capabilities to knowledge-intensive tasks through reasoning-driven retrieval, often referred to as Agentic RAG, in which the model decides when to retrieve external knowledge and what information to seek during reasoning (Li et al., 2025b,c; Liang et al., 2025; Ravuru et al., 2024). By coordinating retrieval with intermediate reasoning states, Agentic RAG can support complex questions requiring multi-hop inference or evidence aggregation, settings in which standard retrieve-then-read RAG may produce fragmented or contradictory responses (Liang et al., 2025).

However, this tight coupling between reasoning states and retrieval decisions introduces substantial operational overhead, reflecting the broader challenge of efficient exploration in sequential decision-making agents (Arumugam and Griffiths, 2026). As shown in Figure 1, a strong Agentic RAG baseline (Li et al., 2025b) often produces incorrect trajectories that are longer and more repetitive, repeatedly issuing near-duplicate retrieval queries without escaping an unproductive search direction. Although interleaved reasoning and retrieval can support complex evidence seeking (Jin et al., 2025; Li et al., 2025c), they can also lead to excessive token use, redundant reasoning steps (Han et al., 2025; Sui et al., 2025), reasoning derailment (Kim et al., 2024a), and rumination-like over-thinking, where the model revisits similar problem formulations or retrieval directions without progress (Marjanović et al., 2026; Sui et al., 2025).

At the same time, simply truncating the trajectory is insufficient: complex reasoning often requires intermediate steps (Wei et al., 2022), while overly aggressive control may cause under-thinking by prematurely abandoning useful reasoning or retrieval directions (Wang et al., 2025). Thus, efficient Agentic RAG should be defined not by shorter trajectories alone, but by adaptive control over evidence seeking: encouraging continued exploration when the evidence state is under-developed, while curtailing redundant search once sufficient evidence has been obtained.

To address these challenges, we introduce RE-GUIDE, a guided Agentic RAG framework that im-

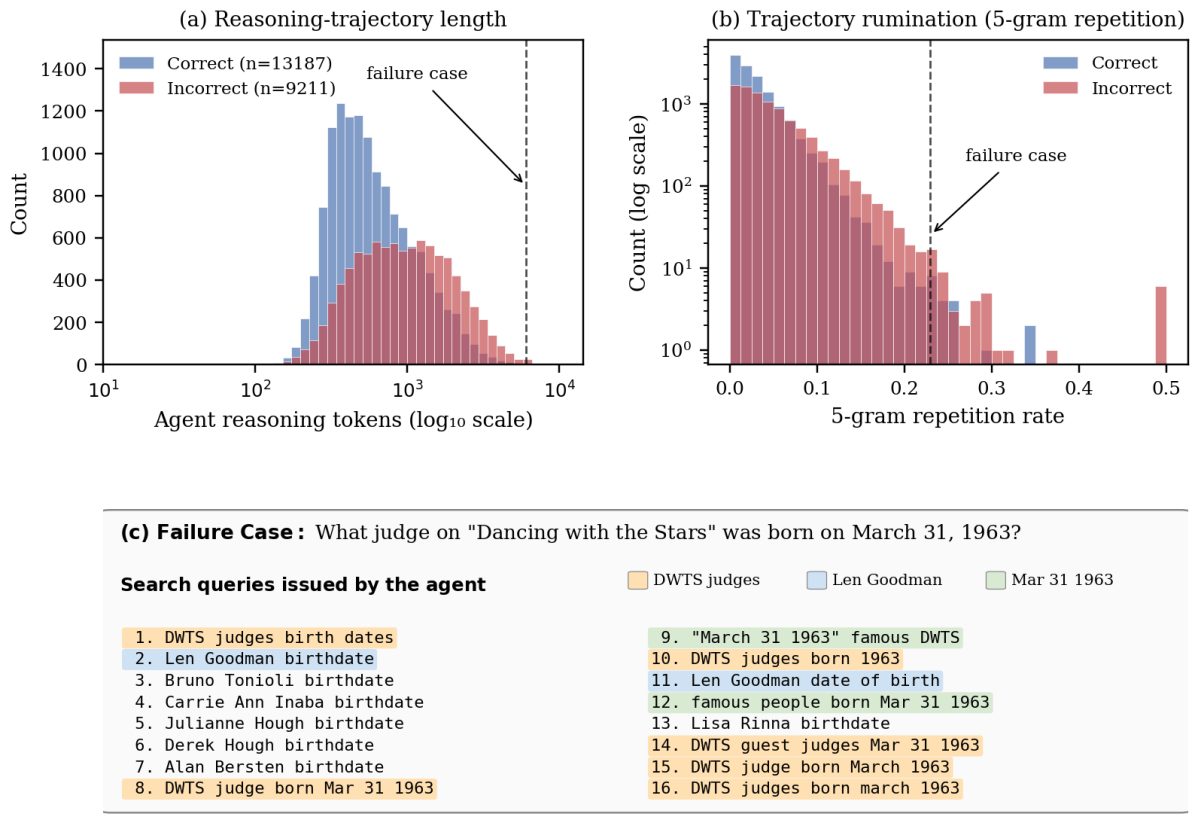

Figure 1: **Failure modes of long reasoning trajectories in Agentic RAG.** For Search-o1 (QwQ-32B) on HotpotQA, 2WikiMultihopQA, and MuSiQue, incorrect predictions exhibit heavier tails in trajectory length (a) and 5-gram repetition rate (b). Dashed lines mark the failure case (c), where the agent issues near-duplicate searches and remains trapped in an unproductive retrieval direction. Search queries in (c) are abbreviated for readability (e.g., DWTS).

proves agent-level reasoning efficiency through two inference-time guidance modules. RETRIEVAL-GUIDE decomposes implicit information needs into concrete retrieval targets before search, while REASONING-GUIDE uses post-retrieval evidence-sufficiency feedback to regulate answering versus continued exploration under a controlled budget.

In summary, our contributions are as follows:

(1) We conduct a behavioral analysis of evidence-seeking trajectories in reasoning-model-based Agentic RAG, identifying two opposite inefficiency patterns: under-thinking, where agents fail to ground uncertain reasoning through sufficient retrieval, and over-thinking, where agents repeatedly pursue redundant or drifting retrieval directions.

(2) We propose RE-GUIDE, an inference-time framework that structures sequential evidence seeking through pre-retrieval planning (RETRIEVAL-GUIDE) and post-retrieval exploration regulation (REASONING-GUIDE), allowing the agent to adapt its retrieval and reasoning trajectory to the evolving evidence state.

(3) Across six QA evaluations and three reasoning models, RE-GUIDE on average improves F1 by 3.4% and reduces agent-generated tokens by 27.2%, achieving a 35.4% gain in F1 per agent-generated token; further analyses show that these gains stem from adaptive regulation rather than uniform trajectory shortening.

2 Related Work

2.1 Efficient Reasoning

Efficient reasoning aims to allocate computation effectively by reducing redundant intermediate reasoning while preserving the steps needed to solve a problem (Qu et al., 2025). Existing methods address this issue through both training-time and inference-time interventions. Training-time approaches encourage models to generate concise or controllable reasoning traces via fine-tuning or reward modeling (Ma et al., 2025; Sel et al., 2025), whereas inference-time approaches reallocate test-time compute without modifying model parameters. Representative methods across these lines of work span dynamic length budgeting (Han et al., 2025;

Lin et al., 2026), adaptive fast–slow reasoning (Su et al., 2025), cost-aware model routing (Ong et al., 2025), multi-path reasoning or search (Wang et al., 2023; Yao et al., 2023a), and early pruning for multi-sample inference (Sun et al., 2024).

Efficiency becomes more challenging in agentic settings, where computation includes planning, tool calls, environment interaction, and external knowledge access. Prior work reduces such overhead by planning function-call dependencies and executing independent calls in parallel (Kim et al., 2024b) or decoupling multi-step reasoning from tool-based knowledge instantiation (Gao et al., 2025).

However, retrieval-augmented agents introduce an additional source of inefficiency: unproductive evidence seeking. Existing efficiency-oriented methods mainly shorten reasoning traces or optimize tool execution, but they do not explicitly regulate whether a retrieval-augmented agent should continue searching, revise its retrieval direction, or stop and answer. RE-GUIDE addresses this gap by guiding both retrieval planning and post-retrieval reasoning in Agentic RAG.

2.2 Agentic RAG

Agentic RAG extends retrieval-augmented generation by interleaving intermediate reasoning with external knowledge access, building on reasoning-and-acting frameworks such as ReAct (Yao et al., 2023b) and active or interleaved retrieval methods such as IRCoT (Trivedi et al., 2023) and FLARE (Jiang et al., 2023).

Subsequent work improves retrieval control: DRAGIN (Su et al., 2024) decides when and what to retrieve from real-time information needs, Self-RAG (Asai et al., 2024) uses reflection tokens for retrieval and self-critique, and Adaptive-RAG (Jeong et al., 2024) selects retrieval strategies by query complexity. Recent reasoning-model-based systems such as Search-o1 (Li et al., 2025b) and WebThinker (Li et al., 2025c) further integrate search and information gathering into long-form reasoning.

Despite these advances, many methods still rely on the model's evolving reasoning trajectory to regulate retrieval. This dependence is problematic because reasoning models may have limited awareness of their own knowledge boundaries (Zeng et al., 2025), while ungrounded extended reasoning can fail to mitigate, or even exacerbate, hallucinations in knowledge-intensive tasks (Zhao et al., 2025). These limitations motivate RE-GUIDE, which provides explicit guidance for evidence seeking and post-retrieval reasoning control.

3 Problem Formulation

We formulate reasoning-model-based Agentic RAG as a sequential evidence-seeking process. Given a question q and an external corpus $\mathcal{D}$, the agent generates a multi-step trajectory $\tau = (\tau_1, \ldots, \tau_T)$ and produces a final answer $\hat{y}$. To control prompt growth, the agent operates on a selectively retained context. Let $\mathcal{W}_t \subseteq \{1, \ldots, t-1\}$ denote the active index set of prior reasoning steps retained in the prompt, and let $\tilde{\tau}_{\mathcal{W}_t}$ denote the corresponding retained reasoning history ordered chronologically. At step t, the agent conditions on a retained prompt context consisting of $\tilde{\tau}_{\mathcal{W}_t}$, the pre-retrieval guide g^{ret}, and the extracted evidence observations $\{x_i\}_{i \in \mathcal{W}_t}$ and post-retrieval guides $\{g_i^{\text{reas}}\}_{i \in \mathcal{W}_t}$ appended during previous steps. We use a belief-state view to abstract the agent's prompt-conditioned predictive state, rather than defining an explicitly estimated latent variable:

$$b_t(y) = p(y \mid q, \tilde{\tau}_{\mathcal{W}_t}, g^{\text{ret}}, \{x_i\}_{i \in \mathcal{W}_t}, \{g_i^{\text{reas}}\}_{i \in \mathcal{W}_t}). \tag{1}$$

Motivated by Bayesian interpretations of in-context learning (Xie et al., 2022), we view retrieved observations and guide messages as contextual evidence that can shift this predictive state.

At each step, the agent's behavior is abstracted using the action set

$$\mathcal{A} = \{\text{PLAN}, \text{SEARCH}, \text{ANSWER}\}. \tag{2}$$

Since a generated segment may involve multiple operations, we denote the induced step-level action composition as $\mathbf{a}_t = (a_{t,1}, \ldots, a_{t,K_t})$, where each $a_{t,k} \in \mathcal{A}$. Here, PLAN formulates or revises a retrieval intent, SEARCH retrieves evidence from $\mathcal{D}$, and ANSWER produces either an intermediate conclusion or the final answer. RE-GUIDE intervenes at two stages: RETRIEVAL-GUIDE supports the PLAN stage by converting underspecified information needs into concrete retrieval targets, while REASONING-GUIDE uses post-retrieval evidence-sufficiency signals to regulate whether the agent should ANSWER or continue PLAN–SEARCH exploration.

Our objective is to improve answer quality per unit of agent-generated reasoning cost; further formulation details are provided in Appendix B.2.

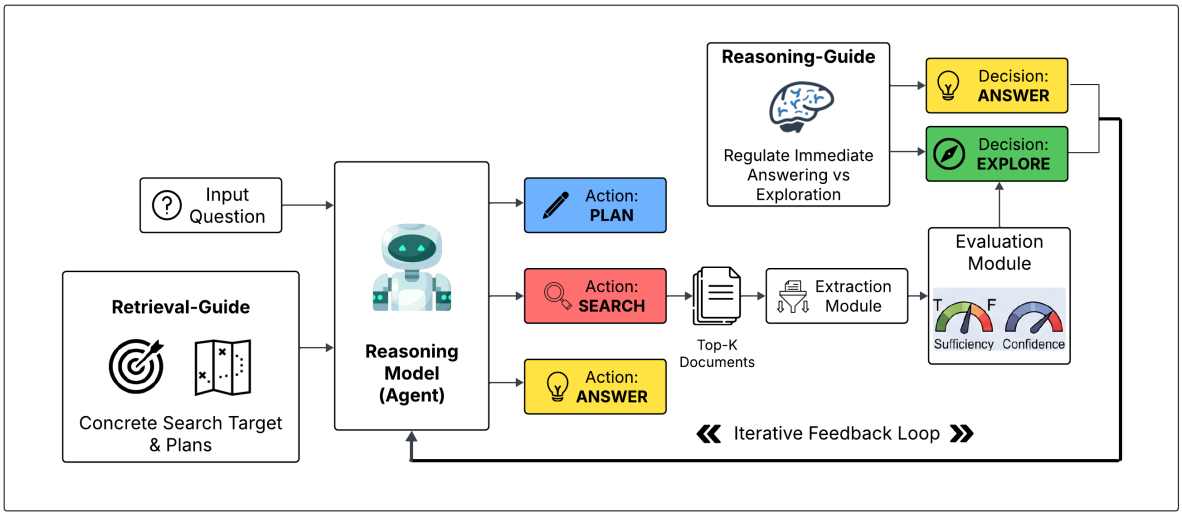

Figure 2: Overview of the RE-GUIDE framework. RETRIEVAL-GUIDE is provided once before iterative reasoning begins to support pre-retrieval planning, while REASONING-GUIDE uses post-retrieval sufficiency feedback to regulate answering versus continued exploration. At each iterative step, the extracted evidence and corresponding REASONING-GUIDE are appended to the next prompt context to guide subsequent reasoning.

4 The RE-GUIDE Framework

Figure 2 illustrates the overall inference workflow of RE-GUIDE. Before iterative reasoning begins, RE-GUIDE inserts a RETRIEVAL-GUIDE into the initial prompt together with the input question. After each retrieval attempt, the extracted evidence x_t and the corresponding REASONING-GUIDE are appended to the next prompt context. To control prompt growth, we retain a bounded recent trajectory window W while preserving intermediate answers that may remain useful for subsequent evidence seeking.

At each reasoning step, the agent autonomously decides whether to invoke search (Li et al., 2025b). When it emits a query z_t enclosed by `<|begin_search_query|>` and `<|end_search_query|>`, the system retrieves a set of top-k passages $E_t = \{d_{t,1}, \ldots, d_{t,k}\}$ from the corpus $\mathcal{D}$ according to the retrieval score $R(z_t, d)$. Following Li et al. (2025b), the Extraction module, implemented with the same underlying reasoning model, converts the retrieved passage set E_t into an evidence observation x_t relevant to the current retrieval intent.

The Evaluation module, also implemented with the same reasoning model but invoked separately, then assesses whether x_t is sufficient for the current retrieval intent. It returns a binary label, SUFFICIENT or INSUFFICIENT, along with a verbalized confidence score on a scale from 0 to 100. We use this verbalized confidence score as a reliability signal, following prior work showing that verbalized confidence can provide useful information for estimating model correctness in aligned and reasoning-oriented LMs (Tian et al., 2023; Zeng et al., 2025).

Rather than relying on the agent's in-generation self-assessment of whether it has gathered sufficient information, we externalize this judgment through a dedicated Evaluation module. This design is motivated by prior work showing that reasoning models may exhibit diminished awareness of their own knowledge boundaries (Zeng et al., 2025).

4.1 Retrieval-Guide

RETRIEVAL-GUIDE provides pre-retrieval guidance for the PLAN stage defined in Section 3. Given a question q, the retrieval-plan generator G_{ret} produces an ordered plan $\rho^{\text{ret}} = (r_1, \ldots, r_m)$, where each step specifies information to find, verify, or compare. The resulting plan is inserted into the agent prompt as optional guidance, allowing the agent to follow or deviate from it as needed.

We fine-tune a Qwen3-8B plan generator on 7K filtered Natural Questions and HotpotQA examples, following prior work on training search agents with high-quality examples (Jiang et al., 2025; Jin et al., 2025). To construct the supervision, we first generate initial plans with dataset-specific reference models and use an LLM-as-a-judge filter (Li et al.,

2025a) to identify hard cases whose plans are insufficiently concrete or poorly targeted. For these selected training examples, GPT-4.1 generates new plans conditioned on the question and training-set annotations, including gold answers and supporting context. Further construction details are provided in Appendix B.5.

4.2 Reasoning-Guide

REASONING-GUIDE provides post-retrieval guidance based on the Evaluation module's sufficiency label and confidence score. When the evidence is judged SUFFICIENT, the guide encourages answer generation, either as an intermediate answer for continued reasoning or as a final answer when the overall information need has been resolved. When the evidence is judged INSUFFICIENT, the guide encourages the agent to revisit the PLAN–SEARCH stages and explore alternative retrieval directions rather than continuing the current insufficient trajectory.

For exploration decisions, we additionally use a confidence-adaptive reasoning budget (Han et al., 2025; Lin et al., 2026). Let $\bar{c}_t = c_t/100$ denote the normalized confidence score of an INSUFFICIENT judgment. We set the prompted exploration budget as

$$B_t = B_{\text{base}} \cdot (\lambda_{\text{bud}} + \bar{c}_t)^r. \tag{3}$$

This design allocates more reasoning budget when the INSUFFICIENT judgment is made with high confidence, treating it as a signal of reducible epistemic uncertainty in the current evidence state (Falck et al., 2024; Lin et al., 2026; Vazhentsev et al., 2023). Conversely, when the insufficiency judgment is uncertain, the budget is limited to avoid over-exploration driven by unreliable evaluator feedback. Hyperparameter details are provided in Appendix B.6.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate RE-GUIDE on six open-domain QA evaluation settings, covering single-hop, ambiguous, and multi-hop QA. For single-hop and ambiguous QA, we use Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and AmbigQA (Min et al., 2020); for multi-hop QA, we use HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022).

Retrieval Corpus and Retriever. All methods use Wikipedia-2018 (Karpukhin et al., 2020) with E5-base-v2 (Wang et al., 2022), ensuring the same retrieval environment.

Reasoning Models. We evaluate RE-GUIDE with three open-weight reasoning models of different scales: DeepSeek-R1-Distill-Llama-8B, DeepSeek-R1-Distill-Qwen-14B (Guo et al., 2025), and QwQ-32B (Qwen Team, 2025).

Evaluation Metrics. Following prior studies (Shi et al., 2026; Yan et al., 2025), we report F1, the average number of agent-generated tokens, and F1 per agent-generated token:

$$\text{F1/AgentTok.} = \frac{\text{F1}}{\overline{T}_{\text{agent}}} \times 10^3, \tag{4}$$

where $\overline{T}_{\text{agent}}$ denotes the average number of reasoning tokens generated by the agent, excluding tokens from auxiliary guide, extraction, and evaluation modules. Because RE-GUIDE introduces auxiliary modules, we interpret F1/Agent Tok. as measuring answer quality relative to the agent's own reasoning cost, rather than full system cost.

To quantify rumination-related redundancy, we additionally report 5-gram repetition rate (Rep.) and normalized lexical entropy (Ent.) following Marjanović et al. (2026). Rep. measures the frequency of repeated contiguous 5-token sequences, while Ent. measures the lexical diversity of the reasoning trajectory after normalization for length. Lower Rep. and higher Ent. therefore indicate less ruminative trajectories. Detailed metric definitions are provided in Appendix D.3.

Baseline. We compare against SEARCH-O1 (Li et al., 2025b), a strong Agentic RAG baseline that autonomously searches and appends retrieved evidence to the next context without explicit retrieval planning or post-retrieval guidance.

5.2 Experimental Results

Agent-Level Reasoning Efficiency. Table 1 summarizes dataset-level averaged results and the relative changes of RE-GUIDE over SEARCH-O1. In general, RE-GUIDE improves F1 by $+3.4\%$ on average while reducing agent-generated tokens by 27.2%, resulting in a $+35.4\%$ gain in F1 per agent-generated token. These results indicate that RE-GUIDE improves the efficiency of agent-level reasoning by allowing the agent to use reasoning tokens more effectively.

Dataset	RE-GUIDE / SEARCH-O1 (Δ)				
	F1↑	Agent Tok.↓	F1/Agent Tok.↑	Rep.↓	Ent.↑
NQ	**36.04** / 33.93 (+6.2%)	**472.9** / 639.2 (−26.0%)	**0.746** / 0.519 (+43.7%)	**0.0314** / 0.0327 (−4.0%)	**0.7796** / 0.7599 (+2.6%)
TQA	**62.45** / 61.51 (+1.5%)	**520.0** / 641.4 (−18.9%)	**1.189** / 0.951 (+25.0%)	**0.0327** / 0.0336 (−2.7%)	**0.7746** / 0.7633 (+1.5%)
AQA	**48.17** / 45.81 (+5.2%)	**492.7** / 625.9 (−21.3%)	**0.959** / 0.712 (+34.7%)	0.0328 / **0.0327** (+0.3%)	**0.7773** / 0.7624 (+2.0%)
HQA	**39.87** / 37.77 (+5.6%)	**614.2** / 863.1 (−28.8%)	**0.635** / 0.436 (+45.6%)	**0.0409** / 0.0457 (−10.5%)	**0.7618** / 0.7436 (+2.4%)
2WQA	**41.81** / 41.54 (+0.6%)	**537.7** / 723.1 (−25.6%)	**0.800** / 0.601 (+33.1%)	**0.0372** / 0.0383 (−2.9%)	**0.7664** / 0.7534 (+1.7%)
MSQ	**19.96** / 19.46 (+2.6%)	**838.9** / 1285.7 (−34.8%)	**0.234** / 0.151 (+55.0%)	**0.0603** / 0.0746 (−19.2%)	**0.7340** / 0.7072 (+3.8%)
Avg.	**41.38** / 40.01 (+3.4%)	**579.4** / 796.4 (−27.2%)	**0.761** / 0.562 (+35.4%)	**0.0392** / 0.0429 (−8.6%)	**0.7656** / 0.7483 (+2.3%)

Table 1: **Per-dataset comparison of RE-GUIDE and SEARCH-O1** averaged over three reasoning models. Each cell shows RE-GUIDE / SEARCH-O1 with the relative change in parentheses. NQ, TQA, AQA, HQA, 2WQA, and MSQ denote Natural Questions, TriviaQA, AmbigQA, HotpotQA, 2WikiMultihopQA, and MuSiQue, respectively. Agent Tok., Rep., and Ent. indicate agent-generated tokens, 5-gram repetition rate, and normalized lexical entropy. Bold marks the better value; ↑/↓ indicate metric direction. See Appendix A.1 for full results.

Model	Method	Planning	Extraction	Evaluation	Reason-in-Docs	Agent	Total-out	Latency (ms)
14B	SEARCH-O1	–	–	–	41.7	715.6	757.3	3196
	RE-GUIDE	45.6	174.9	69.0	–	544.2	833.6	2522
32B	SEARCH-O1	–	–	–	1521.7	1279.3	2801.0	17986
	RE-GUIDE	45.6	1532.7	674.3	–	796.4	3049.1	14171

Table 2: **Average generated-token cost and end-to-end latency** across four datasets (NQ, AmbigQA, HotpotQA, MuSiQue). Planning, Extraction, and Evaluation are auxiliary RE-GUIDE modules, while Reason-in-Docs is used by SEARCH-O1. Agent and Total-out denote agent-generated tokens and all generated output tokens, respectively. 14B denotes DeepSeek-R1-Distill Qwen-14B and 32B denotes QwQ-32B. See Appendix A.2 for full results.

RE-GUIDE improves F1/Agent Tok. in all six evaluations, with notable gains in NQ (+43.7%), HotpotQA (+45.6%), and MuSiQue (+55.0%). These improvements are especially pronounced in multi-hop datasets like MuSiQue which also shows the largest token reduction (−34.8%) demonstrating RE-GUIDE's effectiveness in complex evidence-seeking scenarios.

Beyond quantitative agent-level efficiency gains, trajectory-level analysis shows that RE-GUIDE suppresses rumination-like redundancy. On average, it reduces the 5-gram repetition rate by 8.6% while increasing normalized lexical entropy by 2.3%. The reduction in repetition is especially pronounced on multi-hop datasets such as HotpotQA (−10.5%) and MuSiQue (−19.2%), where unguided agentic retrieval is more prone to repetitive or unproductive search loops.

System-Level Output Tokens and Latency. Table 2 reports system-level costs on the controlled latency-measurement subset. Although RE-GUIDE increases total generated output tokens by 10.1% for the 14B model and 8.9% for the 32B model, it reduces end-to-end latency by approximately 21% in both cases. This shows that total generated tokens alone are not a sufficient indicator of runtime cost in Agentic RAG, since RE-GUIDE shifts part of the generation to auxiliary modules while shortening long-form agent decoding over accumulated contexts. Accordingly, we use agent-generated tokens to isolate agent-level reasoning efficiency, while reporting total output tokens and latency as complementary measures of full system overhead.

6 Further Analysis

We provide a detailed analysis of RE-GUIDE by examining how it addresses distinct failure modes and isolating the contribution of each module.

Mitigating Under-thinking and Over-thinking. Table 3 examines whether RE-GUIDE can recover baseline failures that arise from two opposite trajectory patterns. We partition baseline-incorrect examples into *Short-wrong* and *Long-wrong* groups, defined as the shortest and longest 25% of reasoning trajectories within each dataset and model. We use *Short-wrong* as a proxy for under-thinking, where the agent terminates before collecting sufficient evidence, and *Long-wrong* as a proxy for over-thinking, where the agent continues through unnecessarily long or repetitive reasoning loops.

Type	# Wrong Examples	Overall Rec.	Short-wrong				Long-wrong			
			Group Rec.	Base Rep-5	Longer (%)	ΔLen	Group Rec.	Base Rep-5	Longer (%)	ΔLen
Single-hop QA	12161	10.3	12.3	0.024	52.4	+37	10.4	0.060	4.4	−914
Multi-hop QA	22463	8.1	8.4	0.029	50.9	+50	7.8	0.124	5.7	−1570

Table 3: **Behavioral summary of RE-GUIDE on baseline failures.** Results are stratified by hop type and trajectory length across single-hop QA (NQ, AmbigQA) and multi-hop QA (HotpotQA, MuSiQue) using three reasoning models. Short-wrong and Long-wrong denote the shortest and longest 25% of baseline-incorrect trajectories. Overall/Group Rec. are recovery rates, Base Rep-5 is the baseline 5-gram repetition rate, and Longer/ΔLen measure length changes in recovered cases, micro-averaged over models. See Appendix A.3 for full results.

Model	ΔCalls	Δ%Search	Δ1stSup@5	ΔEvSup	ΔNDup
8B	+79.3%	+2.1 pp	+9.3	+6.1	−2.5
14B	−28.0%	−8.6 pp	+1.9	+2.9	−5.4

Table 4: **Behavioral effect of RETRIEVAL-GUIDE**, averaged over AmbigQA, HotpotQA, and MuSiQue. ΔCalls reports the macro-average of dataset-level relative changes in query calls, while Δ%Search reports the absolute search-rate change in percentage points. Other Δ metrics are in pp and compare against the *w/o Ret.-Guide* ablation. Δ1stSup@5, ΔEvSup, and ΔNDup denote first-query support coverage, consumed-evidence support coverage, and near-duplicate query rate, respectively. 8B/14B denote DeepSeek-R1-Distill Llama-8B/Qwen-14B. See Appendix A.4 for detailed metric definitions and full results.

For *Short-wrong* failures, RE-GUIDE tends to recover examples by enabling additional exploration. Its subgroup recovery rate exceeds the overall recovery rate for both single-hop QA (12.3% vs. 10.3%) and multi-hop QA (8.4% vs. 8.1%), suggesting that short failed trajectories often contain recoverable cases that benefit from further evidence seeking. On average, recovered trajectories become longer in both single-hop (+37 tokens) and multi-hop (+50 tokens) QA, indicating that RE-GUIDE supplies the extra reasoning budget needed to continue searching. Together with the low baseline repetition rates in this group, these patterns suggest that many short-wrong failures reflect insufficient exploration rather than ruminative reasoning.

The pattern is different for *Long-wrong* failures. Here, recovery is consistently associated with shorter trajectories across all model scales. Although the subgroup recovery rate is comparable to the overall recovery rate, the recovered cases show large reductions in trajectory length, especially in multi-hop QA (−1570 tokens). These long-wrong examples also have much higher baseline repetition rates, indicating that their extended trajectories often reflect repetitive or drifting reasoning rather than productive deliberation. When RE-GUIDE recovers these cases, it rarely does so by further lengthening the trajectory: only 4.4% of single-hop and 5.7% of multi-hop recovered cases are longer than the baseline. This suggests that RE-GUIDE mitigates over-thinking by recognizing when continued exploration is unlikely to be useful and guiding the agent toward a timely answer.

Overall, the two failure modes show complementary effects. RE-GUIDE does not uniformly shorten or extend reasoning trajectories; instead, it regulates exploration according to the trajectory state, encouraging further search for under-developed reasoning while curtailing over-extended and repetitive reasoning.

RETRIEVAL-GUIDE calibrates search effort and improves evidence targeting. As shown in Table 4, RETRIEVAL-GUIDE dynamically tailors retrieval behavior to the agent's underlying capabilities. For the weaker 8B model, it increases retrieval activity, with query calls rising by +79.3% and search rate by +2.1 pp, helping the agent search when its internal knowledge is under-supported. This additional exploration improves evidence targeting: first-query support coverage improves by +9.3 pp, and consumed-evidence support rises by +6.1 pp.

Conversely, for the stronger 14B model, RETRIEVAL-GUIDE reduces retrieval activity, lowering query calls by 28.0% and search rate by 8.6 pp, while still improving first-query support (+1.9 pp) and consumed-evidence support (+2.9 pp). This demonstrates that RETRIEVAL-GUIDE calibrates retrieval effort according to the model's actual needs rather than simply forcing more searches.

Furthermore, RETRIEVAL-GUIDE reduces query rumination on average, lowering the near-duplicate

Model	Hop	Stop@ Suff	Cont.@ Insuff	ΔOver-Search	ΔStop@ Oracle	ΔPrem.
8B	Single	100.0	82.6	−23.1%	+4.5	−8.5
	Multi	87.1	80.4	−15.4%	−0.5	+3.5
14B	Single	100.0	87.2	−60.8%	+17.0	+1.5
	Multi	84.2	85.7	−39.4%	+18.0	+2.5

Table 5: **Behavioral effect of REASONING-GUIDE on queries with ≥1 search**, split by hop type. *Single* and *Multi* denote single-hop (NQ, AmbigQA) and multi-hop (HotpotQA, MuSiQue) datasets, respectively; 8B and 14B denote Llama-8B and Qwen-14B. Stop@Suff and Cont.@Insuff measure compliance with SUFFICIENT and INSUFFICIENT judgments. ΔOver-Search, ΔStop@Oracle, and ΔPrem. measure extra retrieval steps after, stopping exactly at, and premature answering before the oracle useful step, respectively. Compliance rates are decision-weighted. Oracle-based Δ metrics macro-average raw values within each hop type before comparison with the *w/o Reas.-Guide* ablation, with ΔOver-Search reported as a relative change and others in percentage points (pp). See Appendix A.5 for full details.

Method	F1↑	Agent Tok.↓	F1/Agent Tok.↑	Rep.↓	Ent.↑
SEARCH-O1	25.39	640.6	0.433	0.0503	0.7480
RE-GUIDE	**27.70**	**508.9**	**0.566**	**0.0470**	**0.7613**
w/o Reas.-Guide	27.54	601.8	0.478	0.0543	0.7504

Table 6: **Average ablation results across four datasets and two reasoning models.** Results are averaged over NQ, AmbigQA, HotpotQA, and MuSiQue using DeepSeek-R1-Distill Llama-8B/Qwen-14B. F1 is reported in %; Agent Tok. denotes average agent-generated tokens. See Appendix A.6 for full results.

query rate by 2.5 pp and 5.4 pp for the 8B and 14B models, respectively. Overall, these results suggest that RETRIEVAL-GUIDE calibrates search effort to model needs while improving evidence targeting and reducing redundant queries.

REASONING-GUIDE improves stopping decisions by reducing over-search. Table 5 shows that the agent largely follows the Evaluation module's sufficiency signals: it stops after every SUFFICIENT judgment on single-hop QA for both models, and continues after most INSUFFICIENT judgments across all settings (80–87%).

The oracle-based metrics provide a more targeted assessment of stopping quality. For the stronger Qwen-14B model, REASONING-GUIDE substantially reduces over-search on both single-hop (−60.8%) and multi-hop (−39.4%) QA, while improving oracle-stopping rates by +17.0 and +18.0 pp, respectively. For Llama-8B, it also reduces over-search in both single-hop (−23.1%) and multi-hop (−15.4%) QA; on single-hop QA, this is accompanied by lower premature answering (−8.5 pp) and improved oracle stopping (+4.5 pp). These patterns indicate that REASONING-GUIDE primarily strengthens stopping behavior after useful evidence has been reached, with clearer gains for the larger model.

Although premature answering slightly increases in some settings, including Llama-8B on multi-hop QA (+3.5 pp), the aggregate ablation results show no degradation in answer quality. As shown in Table 6, RE-GUIDE with REASONING-GUIDE maintains comparable F1 while reducing agent-generated tokens from 601.8 to 508.9. It also improves F1/Ag.Tok. from 0.478 to 0.566, lowers the repetition rate from 0.0543 to 0.0470, and increases normalized lexical entropy from 0.7504 to 0.7613. Together, these results suggest that REASONING-GUIDE primarily improves the efficiency side of the accuracy–cost trade-off: it reduces redundant post-retrieval reasoning and stabilizes trajectories while maintaining comparable aggregate F1.

7 Conclusion

In this paper, we present RE-GUIDE, a guided Agentic RAG framework that improves agent-level reasoning efficiency through retrieval planning and post-retrieval control. Our study highlights that a central challenge in Agentic RAG is not merely interleaving reasoning with retrieval, but adaptively regulating evidence seeking through retrieval planning, evidence-sufficiency assessment, and stopping decisions. When reasoning models rely only on their own evolving trajectories for these decisions, evidence seeking can become inefficient, leading either to premature termination or to redundant retrieval loops.

RE-GUIDE addresses this challenge by explicitly guiding evidence seeking before and after retrieval. Across six open-domain QA evaluations and three reasoning models, RE-GUIDE consistently improves answer quality while substantially reducing agent-generated tokens, showing that its gains arise from adaptive trajectory regulation rather than uniform shortening. Overall, our findings suggest that state-aware guidance for retrieval and reasoning may offer a promising direction for improving the agent-level efficiency and reliability of reasoning-model-based Agentic RAG.

8 Limitations

First, our efficiency metrics focus on the agent-generated tokens rather than the total system-level generation cost. Although our latency analysis indicates that the overhead from auxiliary planning and evaluation is typically offset by reduced sequential decoding, the overall cost–latency trade-off may vary across different serving environments and model backends.

Second, the effectiveness of the framework depends partly on the quality of its guidance signals. Faulty question decomposition or miscalibrated sufficiency judgments can induce suboptimal search trajectories or premature answering. Future work could extend this direction by developing calibration and verification mechanisms that further improve retrieval planning and sufficiency estimation across diverse agentic settings.

Third, our evaluation is constrained to open-domain QA using a fixed Wikipedia corpus and a single dense retriever. Future work should explore how these guidance principles generalize to broader agentic environments that incorporate heterogeneous tools, multimodal evidence, or dynamic corpora.

Finally, RE-GUIDE is explicitly tailored to Agentic RAG and evidence-based QA. Generalizing its planning and evaluation mechanisms to other complex agentic tasks such as code generation, mathematical reasoning, or web navigation remains an open question that warrants further investigation.

References

Dilip Arumugam and Thomas L. Griffiths. 2026. Toward efficient exploration by large language model agents. In *International Conference on Learning Representations*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. 2024. Is in-context learning in large language models Bayesian? a martingale perspective. In *International Conference on Machine Learning*.

Silin Gao, Jane Dwivedi-Yu, Ping Yu, Xiaoqing Ellen Tan, Ramakanth Pasunuru, Olga Golovneva, Koustuv Sinha, Asli Celikyilmaz, Antoine Bosselut, and Tianlu Wang. 2025. Efficient tool use with chain-of-abstraction reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2727–2743, Abu Dhabi, UAE. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855, Vienna, Austria. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. s3: You don't need that much data to train a search agent via RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21599–21617, Suzhou, China. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Minsoo Kim, Jongyoon Kim, Jihyuk Kim, and Seungwon Hwang. 2024a. QuBE: Question-based belief enhancement for agentic LLM reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21403–21423, Miami, Florida, USA. Association for Computational Linguistics.

Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024b. An LLM compiler for parallel function calling. In *International Conference on Machine Learning*.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025a. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.

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

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025c. WebThinker: Empowering large reasoning models with deep research capability. In *Advances in Neural Information Processing Systems*.

Jintao Liang, Sugang, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. 2025. Reasoning RAG via system 1 or system 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1954–1966, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. 2026. Plan and budget: Effective and efficient test-time scaling on reasoning large language models. In *International Conference on Learning Representations*.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035, Vienna, Austria. Association for Computational Linguistics.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, and Siva Reddy. 2026. DeepSeek-R1 thoughtology: Let's think about LLM reasoning. *Transactions on Machine Learning Research*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. AIMO-2 winning solution: Building state-of-the-art mathematical reasoning models with OpenMathReasoning dataset. *arXiv preprint*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*.

OpenAI. 2024a. Learning to Reason with LLMs.

OpenAI. 2024b. OpenAI o1 System Card.

Xiaoye Qu, Yafu Li, Zhao-Chen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint*.

Qwen Team. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Chidaksh Ravuru, Sagar Srinivas Sakhinana, and Venkataramana Runkana. 2024. Agentic retrieval-augmented generation for time series analysis. *arXiv preprint*.

Bilgehan Sel, Lifu Huang, Naren Ramakrishnan, Ruoxi Jia, and Ming Jin. 2025. LLMs can reason faster only if we let them. In *International Conference on Machine Learning*.

Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. 2026. SwiReasoning: Switch-thinking in latent and explicit for pareto-superior reasoning LLMs. In *International Conference on Learning Representations*.

Andy DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qinqing Zheng. 2025. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *International Conference on Learning Representations*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. 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)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter L. Bartlett, and Andrea Zanette. 2024. Fast best-of-N decoding via speculative rejection. In *Advances in Neural Information Processing Systems*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Thoughts are all over the place: On the underthinking of o1-like LLMs. In *International Conference on Machine Learning*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*.

JianZhi Yan, Le Liu, Youcheng Pan, Shiwei Chen, Zike Yuan, Yang Xiang, and Buzhou Tang. 2025. From long to lean: Performance-aware and adaptive chain-of-thought compression via multi-round refinement. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12279–12295, Suzhou, China. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Qingcheng Zeng, Weihao Xuan, Leyang Cui, and Rob Voigt. 2025. Thinking out loud: Do reasoning models know when they're right? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1407, Suzhou, China. Association for Computational Linguistics.

Duzhen Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2026. From system 1 to system 2: A survey of reasoning large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3335–3354.

James Xu Zhao, Bryan Hooi, and See-Kiong Ng. 2025. Test-time scaling in reasoning models is not effective for knowledge-intensive tasks yet. *arXiv preprint*.

Appendix

A: Full Experimental Results

- A.1: Agent-Level Reasoning Efficiency
- A.2: Cost and Latency Analysis
- A.3: Under-thinking and Over-thinking Analysis
- A.4: RETRIEVAL-GUIDE Effect
- A.5: REASONING-GUIDE Effect
- A.6: REASONING-GUIDE Ablation

B: Method Details

- B.1: Implementation Details
- B.2: Detailed Problem Formulation
- B.3: Retained Context Construction
- B.4: Module Interfaces
- B.5: RETRIEVAL-GUIDE Construction
- B.6: Reasoning-Budget Configuration
- B.7: Full Inference Procedure
- B.8: Inference Limits
- B.9: Retrieval Configuration
- B.10: Generation Configuration
- B.11: Baseline Adaptation

C: Prompt Templates

- C.1: Main QA Prompt
- C.2: Main QA Input Format
- C.3: Single-hop Demonstration
- C.4: Multi-hop Demonstration
- C.5: Insufficient-Evidence Demonstration
- C.6: Evidence Extraction Prompt
- C.7: Retrieval-Sufficiency Evaluation Prompt
- C.8: Reasoning-Guide Message Templates

D: Evaluation Details

- D.1: Answer Extraction and Normalization
- D.2: Token Accounting
- D.3: Metric Definitions

E: Qualitative Examples

A Full Experimental Results

A.1 Agent-Level Reasoning Efficiency

Table 7 presents the comprehensive evaluation results comparing RE-GUIDE against the SEARCH-O1 baseline across all six open-domain QA evaluations and three reasoning models. This table provides the absolute values for all evaluation metrics-answer accuracy (F1), agent reasoning cost (Ag.Tok.), agent-level reasoning efficiency (F1/Ag.Tok.), and trajectory rumination (Rep. and Ent.) discussed in Section 5.

A.2 Cost and Latency Analysis

As discussed in Section 6, evaluating the operational efficiency of Agentic RAG requires looking beyond the agent's core reasoning tokens to account for the overhead of auxiliary guidance modules and the actual wall-clock latency. Table 8 provides a detailed, per-module breakdown of generated token costs and end-to-end latency across both 14B and 32B models. Further implementation details are provided below in Section B.1.

A.3 Under-thinking and Over-thinking Analysis

Tables 9 and 10 provide the full, per-dataset and per-model breakdown of the failure-mode analysis discussed in Section 6. The tables detail the specific subgroup recovery rates, baseline 5-gram repetition rates, and the subsequent changes in reasoning trajectory length when RE-GUIDE successfully corrects these failures.

A.4 RETRIEVAL-GUIDE Effect

We directly analyze the generated search trajectories of two models (Llama-8B, Qwen-14B) on datasets with supporting-document annotations (AmbigQA, HotpotQA, and MuSiQue). Table 11 provides the full behavioral results supporting the summary presented in the main text.

For each question, we parse the search queries the agent issued from its raw output and re-run the same E5 retriever to obtain the top-5 passages each query would have produced. We then quantify (i) how often the agent retrieves and (ii) how much gold-supporting evidence its retrievals actually surface.

Type	Model	Dataset	RE-GUIDE					SEARCH-O1 (baseline)				
			F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑	F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑
Single-hop QA	Llama 8B	NQ	**23.85**	**424.9**	**0.561**	**0.0366**	**0.7748**	21.68	474.2	0.457	0.0372	0.7625
		TQA	**46.34**	**481.7**	**0.962**	**0.0400**	**0.7660**	44.16	561.1	0.787	0.0423	0.7607
		AQA	**31.51**	**441.3**	**0.714**	**0.0370**	**0.7726**	28.88	498.3	0.580	0.0396	0.7638
	Qwen 14B	NQ	**32.63**	**466.1**	**0.700**	0.0362	**0.7717**	29.51	560.1	0.527	**0.0341**	0.7591
		TQA	**61.81**	**515.4**	**1.199**	0.0390	**0.7678**	60.84	576.4	1.056	**0.0372**	0.7611
		AQA	**43.99**	**497.2**	**0.885**	0.0395	**0.7688**	39.70	543.5	0.731	**0.0338**	0.7604
	QwQ-32B	NQ	**51.63**	**527.8**	**0.978**	**0.0213**	**0.7924**	50.59	883.4	0.573	0.0267	0.7582
		TQA	79.19	**562.9**	**1.407**	**0.0192**	**0.7900**	**79.54**	786.8	1.011	0.0213	0.7682
		AQA	**69.02**	**539.7**	**1.279**	**0.0219**	**0.7904**	68.86	835.9	0.824	0.0247	0.7629
Multi-hop QA	Llama 8B	HQA	**25.55**	**462.8**	**0.552**	**0.0437**	**0.7650**	23.69	572.1	0.414	0.0468	0.7566
		2WQA	28.42	**343.3**	**0.828**	**0.0349**	**0.7792**	**28.81**	431.9	0.667	0.0370	0.7704
		MSQ	**11.26**	**565.1**	**0.199**	**0.0633**	**0.7403**	9.75	717.6	0.136	0.0705	0.7270
	Qwen 14B	HQA	**35.27**	**535.7**	**0.658**	**0.0501**	**0.7607**	32.75	722.9	0.453	0.0524	0.7447
		2WQA	**36.20**	**425.0**	**0.852**	0.0454	**0.7655**	34.71	563.6	0.616	**0.0431**	0.7575
		MSQ	**17.49**	**677.6**	**0.258**	**0.0698**	**0.7366**	17.15	1035.8	0.166	0.0878	0.7098
	QwQ-32B	HQA	**58.78**	**844.2**	**0.696**	**0.0290**	**0.7598**	56.88	1294.2	0.440	0.0379	0.7294
		2WQA	60.81	**844.8**	**0.720**	**0.0312**	**0.7545**	**61.11**	1173.8	0.521	0.0348	0.7324
		MSQ	31.12	**1273.9**	**0.244**	**0.0477**	**0.7250**	**31.48**	2103.7	0.150	0.0656	0.6848

Table 7: **Full comparison of RE-GUIDE and SEARCH-O1** (baseline) on six QA evaluations across three reasoning models. Datasets are abbreviated as follows: NQ (Natural Questions), TQA (TriviaQA), AQA (AmbigQA), HQA (HotpotQA), 2WQA (2WikiMultihopQA), and MSQ (MuSiQue). Ag.Tok. denotes the average number of reasoning tokens generated by the agent. Rep. and Ent. denote the 5-gram repetition rate and normalized lexical entropy, respectively. Higher is better for F1, F1/Ag.Tok., and Ent.; lower is better for Ag.Tok. and Rep. Bold marks the better value for each metric within each model–dataset pair.

Dataset	Model	Method	Planning	Extraction	Evaluation	Reason-in-Docs	Agent	Total-out	Latency (ms)
Natural Questions	14B	RE-GUIDE	20.6	71.5	30.2	–	466.1	588.4	1406
		SEARCH-O1	–	–	–	14.9	560.1	575.0	1800
	32B	RE-GUIDE	20.6	1095.4	438.8	–	527.8	2082.6	9020
		SEARCH-O1	–	–	–	1126.2	883.4	2009.6	13466
AmbigQA	14B	RE-GUIDE	42.4	97.4	45.0	–	497.2	682.1	2074
		SEARCH-O1	–	–	–	11.1	543.5	554.6	2588
	32B	RE-GUIDE	42.4	1011.7	424.6	–	539.7	2018.4	9806
		SEARCH-O1	–	–	–	1075.8	835.9	1911.7	11438
HotpotQA	14B	RE-GUIDE	56.4	229.6	95.4	–	535.7	917.1	2760
		SEARCH-O1	–	–	–	51.8	722.9	774.8	3149
	32B	RE-GUIDE	56.4	1783.3	814.2	–	844.2	3498.1	15459
		SEARCH-O1	–	–	–	1735.2	1294.2	3029.4	21105
MuSiQue	14B	RE-GUIDE	63.1	300.9	105.3	–	677.6	1146.9	3847
		SEARCH-O1	–	–	–	89.1	1035.8	1124.8	5247
	32B	RE-GUIDE	63.1	2240.5	1019.6	–	1273.9	4597.2	22397
		SEARCH-O1	–	–	–	2149.7	2103.7	4253.4	25936

Table 8: **Per-module generated-token cost and end-to-end latency**, reported for both 14B and 32B models. Only generated tokens are counted. Dashes indicate modules absent from the corresponding method. LATENCY reports the average end-to-end wall-clock time per question (ms). The planner contributes 25.3 ms per question on average across the four datasets.

Data	Model	Wrong items	Overall Recovery%	Short-wrong: shortest 25% failures				
				Subgroup Recovery%	Base Rep-5	Longer on Recovered %	Mean ΔLen on Recovered	n
Single-hop QA								
NQ	8B	3122	7.1	8.8	0.029	58.0	+37	783
NQ	14B	2911	9.9	10.8	0.026	59.5	+64	729
NQ	32B	2226	10.0	12.0	0.015	32.8	−11	557
AQA	8B	1611	10.3	12.9	0.028	59.6	+34	403
AQA	14B	1441	16.4	20.7	0.025	58.7	+84	363
AQA	32B	850	14.4	15.0	0.017	37.5	−38	214
Multi-hop QA								
HQA	8B	6216	7.8	8.7	0.029	60.7	+51	1555
HQA	14B	5652	10.2	11.7	0.032	52.7	+34	1422
HQA	32B	4125	11.7	10.0	0.015	32.0	+5	1032
MSQ	8B	2329	2.9	3.0	0.040	27.8	−27	591
MSQ	14B	2210	4.5	4.5	0.041	48.0	+37	553
MSQ	32B	1931	5.8	5.6	0.023	81.5	+377	483
Micro-average by hop type								
Single-hop	—	12161	10.3	12.3	0.024	52.4	+37	3049
Multi-hop	—	22463	8.1	8.4	0.029	50.9	+50	5636

Table 9: **Under-thinking mitigation behavior of RE-GUIDE**. We split baseline-incorrect questions by baseline reasoning length (in tokens) within each dataset and model. Datasets are abbreviated as follows: NQ (Natural Questions), AQA (AmbigQA), HQA (HotpotQA), and MSQ (MuSiQue). Short-wrong denotes the shortest 25% of failures, used as a proxy for under-thinking. Overall Recovery% is the overall recovery rate, whereas Subgroup Recovery% is computed within this specific subgroup. Base Rep-5 is the baseline 5-gram repetition rate; Longer on Recovered % and Mean ΔLen (in tokens) on Recovered are measured only on successfully recovered cases.

Data	Model	Wrong items	Overall Recovery%	Long-wrong: longest 25% failures				
				Subgroup Recovery%	Base Rep-5	Longer on Recovered %	Mean ΔLen on Recovered	n
Single-hop QA								
NQ	8B	3122	7.1	5.6	0.054	6.8	−420	786
NQ	14B	2911	9.9	10.5	0.053	2.6	−489	733
NQ	32B	2226	10.0	12.9	0.061	5.6	−1286	557
AQA	8B	1611	10.3	8.7	0.074	5.7	−1170	404
AQA	14B	1441	16.4	15.0	0.068	3.7	−862	361
AQA	32B	850	14.4	16.4	0.061	2.9	−1534	213
Multi-hop QA								
HQA	8B	6216	7.8	7.7	0.098	7.6	−575	1555
HQA	14B	5652	10.2	8.3	0.125	5.9	−1284	1414
HQA	32B	4125	11.7	13.0	0.102	3.7	−2383	1032
MSQ	8B	2329	2.9	1.5	0.148	0.0	−1133	583
MSQ	14B	2210	4.5	4.9	0.203	11.1	−1822	554
MSQ	32B	1931	5.8	6.2	0.131	3.3	−2914	483
Micro-average by hop type								
Single-hop	—	12161	10.3	10.4	0.060	4.4	−914	3054
Multi-hop	—	22463	8.1	7.8	0.124	5.7	−1570	5621

Table 10: **Over-thinking mitigation behavior of RE-GUIDE.** We split baseline-incorrect questions by baseline reasoning length (in tokens) within each dataset and model. Datasets are abbreviated as follows: NQ (Natural Questions), AQA (AmbigQA), HQA (HotpotQA), and MSQ (MuSiQue). Long-wrong denotes the longest 25% of failures, used as a proxy for over-thinking. Overall Recovery% is the overall recovery rate, whereas Subgroup Recovery% is computed within this specific subgroup. Base Rep-5 is the baseline 5-gram repetition rate; Longer on Recovered % and Mean ΔLen (in tokens) on Recovered are measured only on successfully recovered cases.

Dataset	Method	*Retrieval activity*		*Rumination*	*Support coverage*	
		Calls	%Search	NDup↓	1stSup@5↑	EvSup↑
Llama-8B						
AmbigQA	RE-GUIDE	0.03	2.8	55.6	89.1	61.8
	w/o Ret.-Guide	0.05	3.3	83.3	80.0	61.5
HotpotQA	RE-GUIDE	0.10	4.0	39.7	66.3	74.5
	w/o Ret.-Guide	0.05	2.0	35.3	62.2	67.6
MuSiQue	RE-GUIDE	0.25	8.4	34.7	59.8	64.7
	w/o Ret.-Guide	0.09	3.5	18.9	45.2	53.6
Qwen-14B						
AmbigQA	RE-GUIDE	0.25	13.5	38.5	82.3	66.4
	w/o Ret.-Guide	0.30	17.1	50.6	76.7	59.8
HotpotQA	RE-GUIDE	0.57	21.0	20.2	64.6	81.7
	w/o Ret.-Guide	0.85	31.2	21.7	64.0	81.5
MuSiQue	RE-GUIDE	0.67	21.6	26.4	46.8	61.0
	w/o Ret.-Guide	1.02	33.6	29.0	47.2	59.0

Table 11: **Retrieval-behavior analysis of RETRIEVAL-GUIDE** on three datasets with supporting-document annotations (AmbigQA, HotpotQA, and MuSiQue) and two different models. We compare the full RE-GUIDE framework against an ablation variant without the pre-retrieval guide (*w/o Ret.-Guide*). NDup (near-duplicate rate) diagnoses query rumination. All percentage entries are scaled to $[0, 100]$.

Model	Dataset	Stop after Suff		Cont. after Insuff	
		rate (%)	n	rate (%)	n
Llama-8B	Natural Questions	100.0	115	84.1	126
	AmbigQA	100.0	50	72.2	18
	HotpotQA	89.4	218	79.9	483
	MuSiQue	83.5	139	81.0	458
Qwen-14B	Natural Questions	100.0	273	87.8	393
	AmbigQA	100.0	234	86.4	265
	HotpotQA	85.4	1373	86.6	2732
	MuSiQue	80.0	405	83.6	1192

Table 12: **Agent-level compliance with REASONING-GUIDE** on four datasets (NQ, AmbigQA, HotpotQA, MuSiQue) and two different models. This table evaluates how effectively the agent adapts its post-retrieval evidence-seeking behavior based on the explicit sufficiency feedback provided by the Evaluation module. All numbers are computed on trajectories from the full RE-GUIDE pipeline. n denotes the total number of labeled decisions within each category.

Model	Dataset	Mean steps		Over-search↓		Stop@Oracle↑		Premature↓	
		RE-GUIDE	w/o	RE-GUIDE	w/o	RE-GUIDE	w/o	RE-GUIDE	w/o
Llama-8B	Natural Questions	1.88	1.60	0.17	0.05	0.93	0.95	0.38	0.60
	AmbigQA	1.25	1.21	0.03	0.21	0.97	0.86	0.31	0.26
	HotpotQA	2.43	2.63	0.57	0.64	0.67	0.63	0.43	0.38
	MuSiQue	2.98	3.32	0.58	0.72	0.60	0.65	0.74	0.72
Qwen-14B	Natural Questions	2.11	2.64	0.55	0.69	0.86	0.75	0.35	0.35
	AmbigQA	1.86	3.13	0.34	1.58	0.89	0.66	0.28	0.25
	HotpotQA	2.69	3.52	0.80	1.46	0.58	0.37	0.32	0.30
	MuSiQue	3.09	3.70	0.74	1.08	0.61	0.46	0.69	0.66

Table 13: **Detailed behavioral comparison of post-retrieval stopping decisions on questions involving at least one search.** This table evaluates how the REASONING-GUIDE influences the agent's search efficiency and stopping accuracy by contrasting the full RE-GUIDE pipeline against the *w/o Reas.-Guide* ablation, in which the Reasoning-Guide module is removed. In *w/o Reas.-Guide*, both the Evaluation module and the resulting REASONING-GUIDE message are removed. *Over-search* and *Stop@Oracle* are computed exclusively for cases where an oracle useful step is successfully reached.

Behavioral Metrics. We define the following retrieval-trajectory metrics, covering both how often the agent searches and whether its searches surface gold-supporting evidence:

- **Calls**: Average number of search queries issued per question over the evaluation set.
- **%Search**: Fraction of questions for which the agent invokes search at least once.
- **NDup**: Near-duplicate rate, defined as the fraction of multi-query trajectories that contain at least one pair of generated queries with token Jaccard similarity ≥ 0.7. This metric captures query rumination, where the model repeatedly issues highly similar queries instead of exploring distinct search intents.
- **1stSup@5**: Among questions for which the agent issues at least one search query, the fraction whose *first* query retrieves a gold supporting title in its top-5 passages.
- **EvSup**: Among questions for which the agent issues at least one search query, the fraction whose post-extraction evidence text, i.e., the information actually consumed by the agent, mentions a gold supporting title.

Higher NDup on Llama-8B multi-hop reflects task structure, not rumination. For Llama-8B, NDup rises on HotpotQA ($35.3 \rightarrow 39.7$) and MuSiQue ($18.9 \rightarrow 34.7$), despite dropping sharply on AmbigQA ($83.3 \rightarrow 55.6$). This increase is partly a denominator effect. Without RETRIEVAL-GUIDE, Llama-8B rarely searches (e.g., 2.0% on HotpotQA), mostly avoiding queries that require multi-step reasoning. RETRIEVAL-GUIDE doubles this search rate, pushing the model to tackle harder multi-hop questions it previously skipped.

These complex questions structurally require entity-bridged queries, naturally pushing token Jaccard similarity above 0.7 even when targeting distinct facts. Conversely, AmbigQA lacks this bridging structure, allowing the reduction in query rumination to manifest directly as an NDup drop.

Crucially, these lexically overlapping queries are highly productive. Support coverage (e.g., 1stSup@5 and EvSup) improves substantially on both HotpotQA and MuSiQue. If the added queries were genuinely redundant, coverage would stagnate. Thus, the higher NDup reflects the inherent lexical nature of multi-hop search, not unproductive repetition.

A.5 REASONING-GUIDE Effect

Tables 12 and 13 provide a detailed breakdown of this behavior across the evaluated datasets (NQ, AmbigQA, HotpotQA, and MuSiQue) and models (Llama-8B and Qwen-14B).

For each agent step, we extract from the raw output the issued query, the post-extraction evidence text, and the Evaluation-module label (SUFFICIENT or INSUFFICIENT).

Behavioral Metrics. We define an *oracle useful step* using a strict evidence-coverage criterion. For multi-hop datasets, the oracle step is the first retrieval step at which the accumulated evidence covers all gold supporting titles, unless the current step already contains a gold answer string. For single-hop datasets, the current evidence must contain a gold answer string.

We assess post-retrieval behavior using metrics that capture both judgment-based compliance and oracle-based stopping:

- **Stop after Suff**: Fraction of SUFFICIENT judgments after which the agent issues no further search queries and directly generates the final answer.
- **Cont. after Insuff**: Fraction of INSUFFICIENT judgments after which the agent issues at least one additional search query.
- **Over-search Depth**: Average number of additional retrieval steps issued after the oracle stop step.
- **Stop-at-Oracle**: Fraction of questions for which the agent performs no further search beyond the oracle stop step.
- **Premature Answer**: Fraction of questions that produce a final answer before reaching the oracle stop step.

A.6 REASONING-GUIDE Ablation

As discussed in Section 6, the REASONING-GUIDE is responsible for explicitly regulating the agent's post-retrieval decisions. To isolate its contribution, we evaluate a variant of our framework (*w/o Reas.-Guide*) where the Evaluation module and the subsequent sufficiency-based feedback messages are entirely removed. Table 14 provides the detailed, per-dataset breakdown of this ablation study across

Natural Questions

Method	F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑
Llama-8B					
SEARCH-O1	21.68	474.2	0.457	0.0372	0.7625
RE-GUIDE	23.85	424.9	0.561	0.0366	0.7748
w/o Reas.-Guide	23.69	487.3	0.486	0.0383	0.7627
Qwen-14B					
SEARCH-O1	29.51	560.1	0.527	0.0341	0.7591
RE-GUIDE	32.63	466.1	0.700	0.0362	0.7717
w/o Reas.-Guide	31.87	545.1	0.585	0.0414	0.7631

AmbigQA

Method	F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑
Llama-8B					
SEARCH-O1	28.88	498.3	0.580	0.0396	0.7638
RE-GUIDE	31.51	441.3	0.714	0.0370	0.7726
w/o Reas.-Guide	30.81	502.1	0.614	0.0424	0.7613
Qwen-14B					
SEARCH-O1	39.70	543.5	0.731	0.0338	0.7604
RE-GUIDE	43.99	497.2	0.885	0.0395	0.7688
w/o Reas.-Guide	42.20	594.3	0.710	0.0455	0.7580

HotpotQA

Method	F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑
Llama-8B					
SEARCH-O1	23.69	572.1	0.414	0.0468	0.7566
RE-GUIDE	25.55	462.8	0.552	0.0437	0.7650
w/o Reas.-Guide	26.45	548.3	0.482	0.0514	0.7542
Qwen-14B					
SEARCH-O1	32.75	722.9	0.453	0.0524	0.7447
RE-GUIDE	35.27	535.7	0.658	0.0501	0.7607
w/o Reas.-Guide	35.28	640.5	0.551	0.0583	0.7499

MuSiQue

Method	F1↑	Ag.Tok.↓	F1/Ag.Tok.↑	Rep.↓	Ent.↑
Llama-8B					
SEARCH-O1	9.75	717.6	0.136	0.0705	0.7270
RE-GUIDE	11.26	565.1	0.199	0.0633	0.7403
w/o Reas.-Guide	12.36	689.7	0.179	0.0766	0.7294
Qwen-14B					
SEARCH-O1	17.15	1035.8	0.166	0.0878	0.7098
RE-GUIDE	17.49	677.6	0.258	0.0698	0.7366
w/o Reas.-Guide	17.66	807.3	0.219	0.0804	0.7245

Table 14: **Ablation results** on the 8B and 14B reasoning models across four representative QA datasets. F1 is reported in %; Ag.Tok. denotes average agent reasoning tokens per question; Rep. and Ent. denote 5-gram repetition rate and lexical entropy, respectively. In *w/o Reas.-Guide*, both the Evaluation module and the resulting REASONING-GUIDE message are removed. Higher is better for F1, F1/Ag.Tok., and Ent.; lower is better for Ag.Tok. and Rep.

four representative datasets (NQ, AmbigQA, HotpotQA, and MuSiQue) and two reasoning models (Llama-8B and Qwen-14B).

B Method Details

B.1 Implementation Details

Experiments were conducted on NVIDIA RTX 8000 (Turing, 48GB VRAM) and RTX A6000 (Ampere, 48GB VRAM) GPUs. For latency analysis, we use only the subset of runs measured on RTX 8000 GPUs, ensuring that both SEARCH-O1 and RE-GUIDE are compared under the same hardware environment. Although both GPUs share the same memory capacity, runs measured on RTX A6000 GPUs are excluded to avoid hardware-induced confounds stemming from generational compute differences.

B.2 Detailed Problem Formulation

Building on the belief-state view and the action space $\mathcal{A}$ introduced in Section 3, we provide a more detailed formulation of the agent policy and the efficiency objective used to evaluate RE-GUIDE.

Step-level agent policy. Although the agent is implemented as an autoregressive generator that produces a text segment τ_t at each step, a single segment may correspond to one or more high-level operations in $\mathcal{A}$: PLAN, SEARCH, and ANSWER. We therefore view each segment as inducing a composite step-level action $\mathbf{a}_t = (a_{t,1}, \ldots, a_{t,K_t})$, where $a_{t,k} \in \mathcal{A}$ and K_t denotes the number of abstract operations expressed within the segment. This formulation allows a generation step to represent either a single operation or a structured combination of multiple operations, without assuming a separately trained action-selection module. Formally, we write

$$\mathbf{a}_t \sim \pi_{\text{agent}}\big(\cdot \mid q, \tilde{\tau}_{\mathcal{W}_t}, g^{\text{ret}}, \{x_i\}_{i\in\mathcal{W}_t}, \{g_i^{\text{reas}}\}_{i\in\mathcal{W}_t}\big), \tag{5}$$

where π_{agent} denotes the step-level policy induced by autoregressive generation over composite actions, rather than an explicit action-selection policy.

Belief update. When the composite action $\mathbf{a}_t$ contains a SEARCH operation, the agent expands its belief state using the newly retrieved evidence and its corresponding reasoning guide. This update is not a parametric model update, but a contextual expansion constrained by the active index set. We define the transition as

$$b_{t+1} = \mathcal{U}\big(b_t, x_t, g_t^{\text{reas}}; q, \tilde{\tau}_{\mathcal{W}_{t+1}}, g^{\text{ret}}, \mathcal{W}_{t+1}\big). \tag{6}$$

Here, $\mathcal{U}$ incorporates the parsed evidence x_t and the discrete reasoning guide g_t^{reas} into the context

whenever a search operation is triggered, while the truncation logic determines the next active context set $\mathcal{W}_{t+1}$.

Efficiency objective. Our evaluation objective is to preserve answer quality while reducing the computational cost of the reasoning trajectory. For each instance i, we define the reasoning cost $C(\tau_i)$ as the total number of reasoning tokens generated by the agent over its T_i steps:

$$C(\tau_i) = \sum_{t=1}^{T_i} \ell_{i,t}, \tag{7}$$

where $\ell_{i,t}$ is the number of agent-generated tokens at step t. The average reasoning cost over an evaluation set $\mathcal{E}$ is then

$$\bar{C} = \frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} C(\tau_i). \tag{8}$$

Given a dataset-level quality metric $Q(\mathcal{E})$, such as F1, we define cost-normalized answer efficiency as

$$\mathrm{Eff}(\mathcal{E}) = \frac{Q(\mathcal{E})}{\bar{C}} \times 10^3. \tag{9}$$

This objective corresponds to our main efficiency metric, F1 per agent reasoning token (F1/Agent Tok.). By normalizing answer quality by the number of reasoning tokens generated by the agent itself, the metric captures agent-level efficiency gains rather than improvements caused by arbitrary trajectory truncation.

B.3 Retained Context Construction

For retained context construction, we implement the active index set $\mathcal{W}_t$ introduced in Section 3 by setting the maximum window size to $W = 5$. At step $t > 1$, the active index set $\mathcal{W}_t$ is constructed to include: (i) the initial step index 1, (ii) the W most recent step indices ($\max(1, t - W) \leq i < t$), and (iii) the indices of all prior steps whose trajectories (τ_i) contain intermediate answers.

The retained reasoning history $\tilde{\tau}_{<t}$, as well as the preserved observations $\{x_i\}_{i \in \mathcal{W}_t}$ and reasoning guides $\{g_i^{\text{reas}}\}_{i \in \mathcal{W}_t}$, are constructed by taking the union of these selected steps and preserving their original chronological order.

This design preserves the initial problem interpretation, the agent's recent reasoning context, and intermediate answers that may remain useful for later evidence seeking, while avoiding unbounded prompt growth as the trajectory length increases. We use the same window size $W = 5$ across all datasets and reasoning models.

B.4 Module Interfaces

The Extraction module takes the question q, recent reasoning context, search query z_t, and retrieved passages e_t as input, and outputs an evidence observation x_t. The goal of this module is not to answer the full question, but to extract information relevant to the current retrieval intent.

The Evaluation module takes the question q, recent reasoning context, search query z_t, and evidence observation x_t as input, and outputs a sufficiency label $s_t \in \{\textsc{Sufficient}, \textsc{Insufficient}\}$ and a confidence score $c_t \in [0, 100]$. The sufficiency label indicates whether the extracted evidence satisfies the current information need, while the confidence score reflects the evaluator's confidence in this judgment.

B.5 Retrieval-Guide Construction

Training data construction details. Following the data collection process described in Section 4.1, we use dataset-specific reference models to generate initial plans: `Qwen2.5-7B-Instruct` for NQ and `Qwen2.5-14B-Instruct-GPTQ-Int4` for HotpotQA.

To provide supporting context for the GPT-4.1 teacher, we extract evidence spans from the gold annotations. For NQ, we use the sentence containing the short answer within the annotated long answer. For HotpotQA, we use the annotated supporting sentences from each gold paragraph. In both cases, we expand the extracted evidence with a ± 1-sentence window.

The resulting plans are designed to reflect the complexity of each dataset. NQ plans consist of a single retrieval step, whereas HotpotQA plans contain an ordered sequence of 2–6 retrieval steps. We then apply a secondary LLM-as-a-judge acceptability check, described below, and retain only the candidate data that pass this filtering stage. The teacher prompts are provided in Appendix B.5.

Retrieval-Plan Generator Training. We fine-tune `Qwen3-8B` with QLoRA as the retrieval-plan generator. Based on our preliminary study (detailed below), the final training set consists of 7K filtered examples with a 3:7 mixture of NQ and HotpotQA. The model is trained with completion-only supervision using the retrieval-guide generation prompt provided in Appendix B.5. For each configuration,

we train for three epochs and retain the checkpoint with the lowest validation loss. All configurations use the same random seed for sampling, shuffling, and validation splitting.

Training and test-time conditions. The RETRIEVAL-GUIDE generator is trained only on the Natural Questions and HotpotQA training splits. Gold answers and contexts are used solely to construct offline GPT-4.1 supervision and are never exposed during inference. At test time, a single trained generator takes only the input question to produce an ordered retrieval plan and is used unchanged across all evaluation datasets and reasoning models without further fine-tuning. This plan serves as optional guidance in the initial prompt, leaving the agent fully autonomous to follow, adapt, or ignore it during reasoning.

LLM-as-a-judge. We use dataset-specific LLM judges to assess whether each retrieval plan is sufficiently targeted. Specifically, we use `Qwen2.5-14B-Instruct-GPTQ-Int4` as the judge for NQ (single-hop) and `Qwen2.5-32B-Instruct-GPTQ-Int4` for HotpotQA (multi-hop). The judge receives the question, candidate retrieval-plan, gold supporting context, and gold answer, and returns a binary PASS/FAIL verdict with a brief rationale. A plan receives PASS only if its retrieval steps are specific enough to reliably retrieve the required supporting context and allow the answer to be derived from it. The judge prompts are also provided in Appendix B.5. We report the judge-pass rate as the fraction of evaluation questions whose generated plans receive a PASS verdict.

Preliminary Study for Training Mixture and Scale. We conduct preliminary experiments to determine the training mixture and data scale for constructing the RETRIEVAL-GUIDE. This study does not aim to evaluate the final QA performance of RE-GUIDE; rather, it selects a retrieval-guide training configuration that produces more acceptable retrieval-plans under an LLM-as-a-judge criterion.

Results. To isolate the effect of training mixture and scale, all configurations use greedy decoding and are evaluated on a fixed, disjoint set of hard-selected examples not used during training. Table 15 summarizes the preliminary study. In the mixture-ratio study, we fix the total number of training examples to 1K and vary the NQ-to-HotpotQA ratio. Among the tested ratios, the 3:7 mixture achieves the highest average judge-pass rate. We then fix this ratio and vary the total number of filtered training examples. The 7K setting achieves the highest average judge-pass rate among the tested scales. Therefore, we use 7K filtered examples with a 3:7 NQ-to-HotpotQA ratio to train the retrieval-plan generator in all subsequent experiments.

Setting	NQ	HQA	NQ Pass	HQA Pass	Avg.
Mixture ratio study: total 1K examples					
R1: 3:7	300	700	25.96	26.94	**26.45**
R2: 4:6	400	600	25.64	**27.02**	26.33
R3: 5:5	500	500	**26.48**	23.26	24.87
Scale study: fixed 3:7 mixture					
S1000	300	700	25.96	26.94	26.45
S2000	600	1400	21.66	22.98	22.32
S3000	900	2100	12.32	15.30	13.81
S4000	1200	2800	**27.24**	25.32	26.28
S5000	1500	3500	23.60	25.26	24.43
S6000	1800	4200	11.08	14.26	12.67
S7000	2100	4900	26.26	**27.92**	**27.09**
S8000	2400	5600	16.32	13.32	14.82
S9000	2700	6300	26.98	21.88	24.43
S10000	3000	7000	22.86	22.10	22.48

Table 15: Preliminary construction study for RETRIEVAL-GUIDE. NQ and HQA indicate the number of filtered training examples from Natural Questions and HotpotQA. NQ Pass, HQA Pass, and Avg. report judge-pass rates (%) on a fixed disjoint evaluation set.

Although larger training sets do not always improve the judge-pass rate, we avoid drawing a strong conclusion from this trend because each configuration is evaluated with a single random seed and a fixed judge setup.

Prompts for Retrieval-plan Generator Construction. This section provides the prompts used to train and evaluate the retrieval-plan generator. The system prompt used for the generator during training is identical to the one used at test-time inference.

Retrieval-Plan Generator System Prompt

```
You are an information retrieval planning
    expert. Given a question, generate an
    ordered sequence of concrete retrieval
    steps required to find the answer. Each
    step represents one retrieval action.

Write one step per line, numbered. No other
    text.
```

Teacher Prompt – Single-Hop (NQ)

```
You are an information retrieval planning
    expert.

Given a single-hop question, a supporting
    context, and the answer, generate one
    concrete retrieval step required to
    find the answer.

Output a JSON object with one field:
- "plan": a single retrieval step.

Requirements:
1. The retrieval step must be specific
    enough that a retrieval system would
    reliably return the supporting context.
2. The answer must be confidently derivable
    from the retrieved context.
3. The step must be derivable from the
    question alone. Do not include facts
    that are only available from the answer
    or supporting context.

Question:
{QUESTION}

Supporting context:
{CONTEXT}

Answer:
{ANSWER}
```

Teacher Prompt – Multi-Hop (HotpotQA)

```
You are an information retrieval planning
    expert.

Given a multi-hop question, context
    sources, and the answer, generate an
    ordered sequence of concrete retrieval
    steps required to find the answer.

Output a JSON object with one field:
- "steps": an ordered array of retrieval
    steps.

Requirements:
1. Each step must correspond to one
    concrete retrieval action.
2. The steps must be specific enough that a
    retrieval system would reliably return
    all required context sources.
3. The answer must be confidently derivable
    from the retrieved contexts.
4. Each step must be derivable from the
    question and the results of previous
    steps. Do not include facts that are
    only available from the answer or
    supporting contexts.

Question:
{QUESTION}

Context sources:
{CONTEXTS}

Answer:
{ANSWER}
```

Judge Prompt – Single-Hop (NQ)

```
You are a quality judge for a single-hop
    retrieval plan.

You will be given a question, a predicted
    retrieval plan, the supporting context,
    and the answer.

Pass if the retrieval plan is targeted
    enough that a retrieval system would
    reliably return the supporting context
    and the answer can be confidently
    derived from it.

Fail if the plan is vague, overly broad,
    asks for unrelated information, or does
    not target the answer-bearing
    information.

Respond in exactly two lines:
Line 1: PASS or FAIL
Line 2: One sentence explaining why the
    plan passes or fails.

Question:
{QUESTION}

Predicted retrieval plan:
{PLAN}

Supporting context:
{CONTEXT}

Answer:
{ANSWER}
```

Judge Prompt – Multi-Hop (HotpotQA)

```
You are a quality judge for a multi-hop
    retrieval plan.

You will be given a question, predicted
    retrieval steps, numbered context
    sources, and the answer.

Pass if:
1. The retrieval steps are targeted enough
    that a retrieval system would reliably
    return all context sources needed to
    derive the answer.
2. The steps are ordered so that later
    steps can use information obtained from
    earlier steps.
3. The answer can be confidently derived
    from the retrieved sources.

Fail if the steps are vague, miss a
    required source, use information not
    derivable from the question or previous
    steps, or do not support deriving the
```

```
    answer.

Respond in exactly two lines:
Line 1: PASS or FAIL
Line 2: One sentence explaining why the
    plan passes or fails.

Question:
{QUESTION}

Predicted retrieval steps:
{STEPS}

Context sources:
{CONTEXTS}

Answer:
{ANSWER}
```

B.6 Reasoning-Budget Configuration

We instantiate the adaptive exploration budget in Eq. 3 with $\lambda_{\text{bud}} = 0.6$ and $r = 2$ in our experiments. Unless otherwise stated, the base budget is set to $B_{\text{base}} = 60$ for single-hop QA datasets and $B_{\text{base}} = 90$ for multi-hop QA datasets. For DeepSeek-R1-Distill-Llama-8B, we use a smaller budget scale, setting $B_{\text{base}} = 30$ for single-hop QA datasets and $B_{\text{base}} = 45$ for multi-hop QA datasets.

These values are used as fixed implementation hyperparameters rather than claimed as optimal settings. They are selected through preliminary calibration on a small set of development trajectories. Specifically, we inspected the budget values induced at different insufficiency-confidence levels and adjusted the hyperparameters so that the prompted budget remains large enough to allow corrective reasoning while still preventing uncontrolled exploration. The values are fixed before test evaluation and are not tuned separately for individual test datasets.

B.7 Full Inference Procedure

Algorithm 1 summarizes the inference procedure of RE-GUIDE. It first builds a retrieval guide, then iteratively generates reasoning steps, invokes search when needed, and extracts evidence from retrieved passages. At each search step, evidence sufficiency determines whether the reasoning guide encourages answering or further exploration, and the guide is appended with the evidence to the subsequent prompt context.

Algorithm 1 RE-GUIDE inference procedure.

Require: Question q, corpus $\mathcal{D}$, maximum turns T, maximum searches S, window size W
Ensure: Final answer $\hat{y}$

Generate retrieval plan $\rho^{\text{ret}} \leftarrow G_{\text{ret}}(q)$.
Build retrieval-guide message $g^{\text{ret}} \leftarrow \mathcal{T}_{\text{ret}}(\rho^{\text{ret}})$.
Initialize trajectory $\tau \leftarrow \emptyset$.
Initialize search counter $n \leftarrow 0$.
for $t = 1, \ldots, T$ **do**
  Update active index set $\mathcal{W}_t \leftarrow \{1\} \cup \{\max(1, t - W), \ldots, t-1\}$
    $\cup \{i < t - W \mid \tau_i \text{ has intermediate answer}\}$.
  Construct retained trajectory $\tilde{\tau}_{<t}$ based on $\mathcal{W}_t$.
  Generate $\tau_t \sim \pi_{\text{agent}}\big(\cdot \mid q, \tilde{\tau}_{<t}, g^{\text{ret}}\big)$.
  Append τ_t to τ.
  if τ_t contains a final answer **then**
    return extracted final answer $\hat{y}$.
  else if τ_t invokes search and $n < S$ **then**
    Parse search query z_t from τ_t.
    Set η_t as τ_t excluding z_t to represent the current retrieval intent.
    Retrieve raw passages $e_t \leftarrow \text{TopK}_{d \in \mathcal{D}}(R(z_t, d), k)$.
    Extract evidence observation $x_t \leftarrow M_{\text{ext}}(q, \eta_t, z_t, e_t)$.
    Evaluate sufficiency $(s_t, c_t) \leftarrow M_{\text{eval}}(q, \eta_t, z_t, x_t)$.
    Set $d_t \leftarrow$ ANSWER if $s_t =$ SUFFICIENT; otherwise set $d_t \leftarrow$ EXPLORE.
    if $d_t =$ EXPLORE **then**
      Compute exploration budget B_t using Eq. 3.
      Build reasoning-guide message $g_t^{\text{reas}} \leftarrow \mathcal{T}_{\text{reas}}(d_t, q, c_t, B_t)$.
    else
      Build reasoning-guide message $g_t^{\text{reas}} \leftarrow \mathcal{T}_{\text{reas}}(d_t, q)$.
    end if
    Append evidence observation x_t to τ.
    Append reasoning-guide message g_t^{reas} to τ.
    Update search counter $n \leftarrow n + 1$.
  end if
  if the cumulative sequence length exceeds the predefined limit **then**
    break
  end if
end for
return extracted answer from τ.

B.8 Inference Limits

We use a maximum of $T = 20$ turns and $S = 20$ searches per question. For a fair comparison, the same maximum turn and search limits are applied to both RE-GUIDE and SEARCH-O1.

B.9 Retrieval Configuration

For each generated search query, the retriever returns the top-5 passages for all methods. We use the same top-5 retrieval setting across all methods for fair comparison. Long retrieved documents are truncated to a fixed maximum document length of 3000 before being passed to the extraction module, which keeps extraction prompts within the model context window. Search results are cached by query

string to avoid redundant retrieval calls during evaluation.

B.10 Generation Configuration

We use a fixed decoding configuration across all methods to isolate the effect of the proposed guidance mechanism. For the agent, we use temperature 0.7, top-p 0.8, and top-k sampling 20. The maximum context length is set to 20,480 tokens, and each agent decoding call is limited to 4,096 new tokens. We also apply early termination when the accumulated output becomes abnormally long or when the context limit is reached. Concretely, a sequence is terminated as soon as its accumulated output exceeds 40,000 characters (roughly 6.4K tokens), which serves as a safeguard against degenerate loops and is rarely activated in normal trajectories.

B.11 Baseline Adaptation

SEARCH-O1 was originally designed for web search, whereas our experiments use a fixed Wikipedia corpus with a dense retriever. To adapt the baseline to this corpus-based retrieval setting, we kept the original search markers but added a short instruction encouraging concise keyword-style queries:

```
To perform a search, write a search query
in the format <|begin_search_query|>
your query here <|end_search_query|>,
written as concise keywords only.
```

This instruction aligns SEARCH-O1 with the same prompt-level retrieval interface used by RE-GUIDE.

C Prompt Templates

This section provides the prompt templates used for RE-GUIDE. We include prompts for the agent and auxiliary modules: the retrieval plan generator, the extraction module, and the retrieval-sufficiency evaluator. We also show the reasoning-guide message templates inserted into the agent's trajectory after each retrieval attempt.

C.1 Main QA Prompt

At the beginning of the inference phase, the main QA prompt is provided to the agent along with two demonstrations: one task-specific in-context example matching the hop type of the dataset, and one insufficient-evidence demonstration.

Main QA Instruction

```
You are a question answering assistant with
    the ability to perform searches and
    reason.

You can receive assistance from the system
    guide during the retrieval and
    reasoning process.
You must provide an accurate answer to the
    given question by performing searches
    to retrieve relevant information, and
    reasoning over the retrieved
    information to derive the answer.

To perform a search, write a search query
    in the format <|begin_search_query|>
    your query here <|end_search_query|>,
    written as concise keywords only.
Then, the system will then search for
    relevant content, and return helpful
    information in the format
    <|begin_search_result|> ...extracted
    information... <|end_search_result|>.
The system will also provide guidance on
    whether the search result is
    sufficient, and on the direction your
    reasoning should take next, in the
    format [Reasoning Guide].

The final answer must be in the format
    \boxed{YOUR_ANSWER}.
```

C.2 Main QA Input Format

For each test question, the generated retrieval guide is inserted into the following input format.

Main QA Input Format

```
Question:
{QUESTION}

You may refer to the retrieval guide below
    to inform your search strategy, or feel
    free to take a different approach.

Retrieval Guide:
{RETRIEVAL_GUIDE}
```

C.3 Single-hop Demonstration

Single-hop Demonstration

```
Here is an example of a single-hop question
    that requires one retrieval step:

Question:
who holds the most women's wimbledon titles

You may refer to the retrieval guide below
    to inform your search strategy, or feel
    free to take a different approach.

Retrieval Guide:
```

```
Find the record holder for the most women's
    singles titles at Wimbledon.

Following the retrieval guide, I need to
    find who holds the record for the most
    women's singles titles at Wimbledon.
<|begin_search_query|> most women's
    Wimbledon singles titles record holder
    <|end_search_query|>
<|begin_search_result|> Martina Navratilova
    holds the record for the most women's
    singles titles at Wimbledon, having won
    the championship nine times between
    1978 and 1990. <|end_search_result|>
[Reasoning Guide]: Sufficient.
You now have enough information to answer
    the original question "who holds the
    most women's wimbledon titles". Provide
    the final answer in the format
    \boxed{YOUR_ANSWER}.

The search result clearly states Martina
    Navratilova holds the record with nine
    Wimbledon titles.
\boxed{Martina Navratilova}
```

C.4 Multi-hop Demonstration

Multi-hop Demonstration

```
Here is an example of a multi-hop question
    that requires multiple retrieval steps:

Question:
The Oberoi family is part of a hotel
    company that has a head office in what
    city?

You may refer to the retrieval guide below
    to inform your search strategy, or feel
    free to take a different approach.

Retrieval Guide:
1. Identify the hotel company the Oberoi
    family is part of.
2. Find the city where that hotel company's
    head office is located.

Following step 1 of the retrieval guide, I
    need to identify the hotel company the
    Oberoi family is part of.
<|begin_search_query|> Oberoi family hotel
    company <|end_search_query|>
<|begin_search_result|> The Oberoi family
    is an Indian family involved in the
    hospitality industry. They are the
    founders and owners of The Oberoi
    Group, which operates luxury hotels and
    resorts. <|end_search_result|>
[Reasoning Guide]: Sufficient.
1) If there is still more information to
    retrieve before fully answering the
    original question "The Oberoi family is
    part of a hotel company that has a head
    office in what city?", derive an
    intermediate answer based on the
    retrieved information, then continue
    reasoning toward the next retrieval.
    You may refer to the reasoning context
    above or the retrieval guide to inform
    your search strategy, or feel free to
    take a different approach.
2) If you have sufficient information to
    fully answer the original question,
    provide the final answer in the format
    \boxed{YOUR_ANSWER}.

The Oberoi family is associated with The
    Oberoi Group. I still need to find the
    head office location, so I'll continue
    retrieval.
Following step 2 of the retrieval guide,
    now I need to find where The Oberoi
    Group's head office is located.
<|begin_search_query|> The Oberoi Group
    head office location
    <|end_search_query|>
<|begin_search_result|> The Oberoi Group is
    a hotel company founded in 1934. It
    operates luxury hotels, resorts, and
    cruisers in several countries. The
    company's head office is located in
    Delhi, India. <|end_search_result|>
[Reasoning Guide]: Sufficient.
1) If there is still more information to
    retrieve before fully answering the
    original question "The Oberoi family is
    part of a hotel company that has a head
    office in what city?", derive an
    intermediate answer based on the
    retrieved information, then continue
    reasoning toward the next retrieval.
    You may refer to the reasoning context
    above or the retrieval guide to inform
    your search strategy, or feel free to
    take a different approach.
2) If you have sufficient information to
    fully answer the original question,
    provide the final answer in the format
    \boxed{YOUR_ANSWER}.

The Oberoi family owns The Oberoi Group,
    whose head office is in Delhi.
\boxed{Delhi}
```

C.5 Insufficient-Evidence Demonstration

The prompt also includes a demonstration where the first retrieval result is insufficient. This example teaches the agent to redirect the retrieval instead of repeatedly pursuing the same failed direction.

Insufficient-Evidence Demonstration

```
Here is an example where the first
    retrieval is insufficient and a retry
    with a differently-angled approach is
    needed:

Question:
when did women get the vote in the uk
```

```
You may refer to the retrieval guide below
    to inform your search strategy, or feel
    free to take a different approach.

Retrieval Guide:
Find the year women gained the right to
    vote in the United Kingdom.

Following the retrieval guide, I need to
    find when women gained voting rights in
    the UK.
<|begin_search_query|> UK women vote year
    <|end_search_query|>
<|begin_search_result|> NONE
    <|end_search_result|>
[Reasoning Guide]: Insufficient (confidence
    100%).
Feel free to explore alternative paths,
    such as trying a different search query
    or taking different retrieval steps as
    needed, to derive an answer to the
    question: when did women get the vote
    in the uk. Please think carefully, and
    use up to 153 words.

The retrieval was entirely insufficient, so
    I should switch angles entirely.
    Instead of asking the date directly,
    search for the broader topic of women's
    suffrage history in the UK, which is
    more likely to appear in encyclopedic
    sources.
<|begin_search_query|> women suffrage UK
    history <|end_search_query|>
```

C.6 Evidence Extraction Prompt

Once the documents are retrieved, we guide the extraction module to extract relevant evidence using the prompt below:

Evidence Extraction Prompt

```
You are an extraction assistant.
...

Read the retrieved documents and extract
    information relevant to the current
    information need reflected in the
    search query and recent reasoning, **to
    advance the reasoning toward**
    answering the question.

Rules:
- Use only the retrieved documents.
- Do not speculate or use outside knowledge.
- Keep factual details as written in the
    documents.
- Intermediate bridge facts (e.g., the
    identity of an entity that connects the
    question to a later retrieval step) are
    useful and should be extracted, even
    when they do not directly answer the
    original question.
- Output NONE only when the documents
    contain neither a direct answer nor any
    intermediate fact usable for the next
    reasoning step.

Output format:
- If the retrieved documents contain
    helpful information (direct or
    intermediate) for the current search
    query or recent reasoning, provide the
    information with `**Extracted
    Information**` as shown below.

**Extracted Information**

[Extracted information]

- If the retrieved documents do not contain
    any helpful information (direct or
    intermediate), output the following:

**Extracted Information**

NONE

Question: {QUESTION}
Current Search Query: {SEARCH_QUERY}
Recent Reasoning: {RECENT_REASONING}
Retrieved Documents: {DOCUMENTS}

Now you should read the retrieved documents
    and find helpful information based on
    the current search query
    "{SEARCH_QUERY}" and recent reasoning,
    including any intermediate facts that
    could support the next reasoning step.
```

C.7 Sufficiency Evaluation Prompt

To assess whether the retrieved evidence is sufficient to satisfy the agent's current information need and to determine a confidence score, we use the following evaluation prompt:

Sufficiency Evaluation Prompt

```
You are a retrieval evaluation assistant.

Your task is to determine whether the given
    search result is sufficient to satisfy
    the information need expressed in the
    most recent reasoning to answer the
    question.

Rules:
1. Judge only based on the provided search
    result. Do not infer, speculate, or use
    outside knowledge.
2. Focus only on the information need
    implied by the recent reasoning, not
    the whole question.
3. Output <answer> sufficient </answer>
    only if the search result provides
    enough explicit information to satisfy
    that information need without
    additional search.
4. Output <answer> insufficient </answer>
    if the search result is incomplete,
```

```
    ambiguous, indirect, weakly relevant,
    or missing key information needed to
    satisfy that information need.
5. After that, output a confidence score
    from 0% to 100% indicating how
    confident you are in your
    sufficient/insufficient decision based
    only on the given search result.

Output format:
<answer> sufficient </answer>
<confidence> 90% </confidence>

or

<answer> insufficient </answer>
<confidence> 85% </confidence>

Do not output any explanation, reasoning,
    or extra text.

Question: {QUESTION}

Recent Reasoning: {RECENT_REASONING}

Search Query: {SEARCH_QUERY}

Search Result: {SEARCH_RESULT}
```

C.8 Reasoning-Guide Message Templates

Depending on the sufficiency of the retrieved evidence, we dynamically insert one of the following Reasoning-Guide messages to regulate the agent's next action:

Reasoning-Guide Message: Sufficient Evidence

```
[Reasoning Guide]: Sufficient.
1) If there is still more information to
    retrieve before fully answering the
    original question "{QUESTION}", derive
    an intermediate answer based on the
    retrieved information, then continue
    reasoning toward the next retrieval.
    You may refer to the reasoning context
    above or the retrieval guide to inform
    your search strategy, or feel free to
    take a different approach.
2) If you have sufficient information to
    fully answer the original question,
    provide the final answer in the format
    \boxed{YOUR_ANSWER}.
```

Reasoning-Guide Message: Insufficient Evidence

```
[Reasoning Guide]: Insufficient (confidence
    {CONFIDENCE}\%).
Feel free to explore alternative paths,
    such as trying a different search query
    or taking different retrieval steps as
    needed, to derive an answer to the
    question: {QUESTION}. Please think
    carefully, and use up to {BUDGET} words.
```

D Evaluation Details

D.1 Answer Extraction and Normalization

For QA-style datasets, we extract the final answer from the last occurrence of `\boxed{...}`. We then normalize both prediction and gold answers by lowercasing, removing punctuation and articles, and fixing whitespace. We report token-level F1.

D.2 Token Accounting

We measure agent reasoning cost from its raw generation after removing system-injected content, including retrieval-result blocks delimited by `<|begin_search_result|>...<|end_search_result|>` and reasoning-guide messages starting with `[Reasoning Guide]:`. The filtered text is re-tokenized with the corresponding model tokenizer, and the resulting counts are averaged over questions.

For auxiliary modules, we record generated tokens from the inference runtime for each call, sum them per question, and average the totals over the dataset. All auxiliary counts are measured using the same tokenization procedure as the agent reasoning tokens.

D.3 Metric Definitions

We provide the detailed definitions of the evaluation metrics used in Section 5. For answer quality, we report token-level F1 after applying the answer extraction and normalization procedure described in Appendix D.1.

F1 per Agent Reasoning token. As introduced in Section 5, this metric normalizes answer accuracy by the agent's average reasoning token count. The denominator, $\overline{T}_{\text{agent}}$, is computed strictly according to the token accounting procedure detailed in Appendix D.2.

N-gram Repetition Rate. To measure local surface-level redundancy in reasoning trajectories, we report the 5-gram repetition rate following Marjanović et al. (2026). Let $T = (w_1, \ldots, w_N)$ denote the generated reasoning trajectory of the agent, and let $\mathcal{G}_n(T)$ be the multiset of contiguous n-grams in T. We define the metric as follows:

$$\text{Rep}_n(T) = 1 - \frac{|\text{Unique}(\mathcal{G}_n(T))|}{|\mathcal{G}_n(T)|} \tag{10}$$

In our experiments, we set $n = 5$. A higher Rep_5 indicates that the trajectory repeats more local text spans, reflecting stronger surface-level recurrence.

Normalized Lexical Entropy. We also report normalized lexical entropy following Marjanović et al. (2026). Let V be the set of unique tokens in the trajectory, let $c(w)$ denote the count of token w, and let $p(w) = c(w)/N$. We define the entropy as:

$$\mathrm{Ent}(T) = -\frac{\sum_{w \in V} p(w) \log_2 p(w)}{\log_2 N} \tag{11}$$

Lower entropy indicates that the trajectory is concentrated on a narrower set of tokens, which often reflects repetitive or circular reasoning. Thus, lower Rep_5 and higher $\mathrm{Ent}(T)$ correspond to less ruminative trajectories.

E Qualitative Examples

We provide representative examples illustrating how RE-GUIDE changes the reasoning trajectory.

For readability, we present condensed trajectories that preserve the key reasoning, retrieval, answer, and guide-messages. Each box shows the agent trace and retrieved evidence, whereas the italicized annotation below provides our post-hoc analysis.

Case A. Setup: DeepSeek-R1-Distill-Qwen-14B on HotpotQA (failure mode: *Under-thinking*). **Question:** The film "Arthur Newman" stars Colin Firth and an actress that made her professional debut in what play? *(Gold: The Royal Family.)*

SEARCH-O1 trajectory – Case A (0 retrieval calls, 303 agent's reasoning tokens, F1 = 0.00)

```
[Reason]    Okay, so I need to figure out
    which play the actress who starred with
Colin Firth in ``Arthur Newman'' made her
    professional debut in. First, I should
identify the actress who starred alongside
    Colin Firth. I'm not entirely sure about the
    details, but I can look it up.

[Reason]    Once I know the actress's name,
    the next step is to find out when she
made her professional debut---probably in a
    play or a notable film. Since the
question mentions a play, I should focus on
    her debut in a theater production.

[Reason]    I think the actress in question
    might be someone who has been in other
well-known films or shows, so maybe her
    debut was in a famous play. I'm not
    entirely
certain, so I'll need to do a search to
    confirm.

[Reason]    ``The film `Arthur Newman'
    stars Colin Firth and Rachel Weisz.
    Rachel
Weisz made her professional debut in the
    play `The Beauty Queen of Leenane.' ''

[Answer]    The Beauty Queen of Leenane
    (incorrect, F1 = 0.00)
```

Trace annotation. Although the agent explicitly recognizes the need for verification, no retrieval call is issued. The trajectory instead moves directly from stated uncertainty to an unsupported factual assertion, producing an incorrect cast member and debut play.

RE-GUIDE trajectory – Case A (3 retrieval calls, 1,039 agent's reasoning tokens, F1 = 1.00)

```
[Plan]   (1) Identify the actresses
    starring alongside Colin Firth in
    ``Arthur Newman''.
        (2) Determine which of those
    actresses made her professional debut
    in a specific play.
        (3) Find the title of the play in
    which she made her professional debut.

[Reason]    Recalling from memory, ``Arthur
    Newman'' is a 2012 film. It stars
Colin Firth as King George VI, and I believe
    Helena Bonham Carter plays Queen Elizabeth.
Let me verify the cast through retrieval.

[Search]    actresses in Arthur Newman film
[Result]    Emily Blunt is the actress in
    the film ``Arthur Newman'' alongside
Colin Firth. To determine the play where
    she made her professional debut, further
information is needed.
[Guide]     Insufficient (confidence 95%).
    Feel free to explore alternative paths,
            such as trying a different
    search query or taking other retrieval
            steps as needed, to derive an
    answer to the question [...]. Please
            think carefully, and use up to
    216 words.

[Reason]    Wait, the search result says
    Emily Blunt is in ``Arthur Newman.'' I
thought it was Helena Bonham Carter. I think I
    might have mixed up"Arthur Newman" with
    "The King's
    Speech", where Firth and Bonham Carter
starred together. Let me confirm the cast.

[Search]    cast of Arthur Newman film 2012
[Result]    The film ``Arthur Newman''
    stars Colin Firth and Emily Blunt.
[Guide]     Sufficient. 1) Derive an
    intermediate answer and continue toward
            the next retrieval if more is
    needed. 2) Otherwise, provide the
            final answer in the format
    \boxed{YOUR_ANSWER}.
```

```
[Reason]    Cast confirmed: Colin Firth and
    Emily Blunt. Now, following step
(2)--(3) of the plan, I need to find Emily
    Blunt's professional debut play.

[Search]    Emily Blunt professional debut
    play
[Result]    Emily Blunt made her
    professional debut in the play ``The
    Royal Family''
in November 2001, where she played the role
    of Gwen opposite Dame JudiEnch.
[Guide]     Sufficient. 1) Derive an
    intermediate answer and continue toward
            the next retrieval if more is
    needed. 2) Otherwise, provide the
            final answer in the format
    \boxed{YOUR_ANSWER}.

[Answer]    The Royal Family   (correct, F1
    = 1.00)
```

Trace annotation. RE-GUIDE converts the cast-identification step into explicit retrieval. The first retrieval contradicts the agent's memory-based assumption (Helena Bonham Carter) and triggers a self-correction in which the agent recognizes a confusion with *The King's Speech*; a second retrieval confirms the corrected cast (Emily Blunt), and a third retrieves the debut play required for the final answer.

Observation. This case illustrates an under-thinking failure: SEARCH-O1 recognizes the need for confirmation but makes no retrieval calls, leading to an incorrect cast member and debut play. By contrast, RE-GUIDE turns the same uncertainty into evidence-seeking behavior by verifying the cast, revising its memory-based assumption, and retrieving the debut play. Its longer trajectory, with 1,039 reasoning tokens and three retrieval calls, suggests that RE-GUIDE mitigates premature termination when the evidence state remains under-developed.

Case B. Setup: QwQ-32B on HotpotQA (failure mode: *Over-thinking*).
Question: Bill Woodfull opened pairing with what cricketer who died in April of 1991? *(Gold: Bill Ponsford.)*

SEARCH-O1 trajectory – Case B (20 retrieval calls, 1,957 agent's reasoning tokens, F1 = 0.00)

```
[Reason]    Bill Woodfull was an Australian
    Test captain in the late 1920s and
1930s. I need to find his opening partner
    who died in April 1991. Let me first
identify his opening partners.

[Search]    Bill Woodfull opening partners
    cricket
[Result]    Bill Woodfull's primary opening
    partner was Bill
    Ponsford
(a fellow Victorian and long-term
    Australian opener, including a record
    375-run
stand); other partners include Warren
    Bardsley and Charlie Macartney.

[Reason]    According to the result, the
    primary partner is Ponsford -- but
"Ponsford died in 1983, so that's too
    early." The target must
be a different partner who died in April
    1991.

[Search]    cricketer died april 1991
[Result]    No helpful information found.
    None of the retrieved passages mention
a cricketer who died in April 1991.

[...]       15 further retrieval step(s):
    bill brown cricketer death date;
charlie macartney cricketer death date;
    warren bardsley cricketer death date;
lindsay hassett death date; arthur mailey
    death date; bill woodfull opening
partners first class cricket; cricketer
    died april 1991 (twice more); johnnie
moyes death date; south african cricketer
    died april 1991; cricketer death
april 1991; ernie toshack death date; frank
    tarrant death date; les ames
cricketer death date; alan kippax death date

[Search]    bill woodfull victoria teammates
[Result]    Bill Ponsford was Woodfull's
    primary opening partner for Victoria.
No new candidates emerge.

[Reason]    No luck there. Perhaps the
    cricketer is someone like Bill Johnston.
Let me check his death date.

[Search]    bill johnston cricketer death
    date
[Result]    Bill Johnston died on 25 May
    2007; does not match April 1991, and he
was a bowler rather than an opener.

[Search]    bill woodfull opening partner
    died april 1991
[Result]    No helpful information found.
    Prior searches established Ponsford as
the primary partner but his death year was
    ruled out; no new candidates emerge.

[Reason]    After 20 searches I cannot find
    a cricketer matching both criteria.

[Answer]    (no answer produced)
    (incorrect, F1 = 0.00)
```

Trace annotation. The first retrieval identifies Bill Ponsford as the strongest opening-partner candidate, but the agent rejects him using an unverified memory-based death-date claim. Because the death-date constraint is never checked directly, the subsequent trajectory shifts away from verifying the strongest candidate and toward searching for alternative candidates. This produces a long but unproductive search process that exhausts the retrieval budget.

RE-GUIDE trajectory – Case B (2 retrieval calls, 180 agent's reasoning tokens, F1 = 1.00)

```
[Plan]   (1) Find which cricketer Bill
    Woodfull opened batting pairings with.
        (2) Determine whether that
    cricketer died in April 1991.
        (3) Return the name of the
    cricketer who satisfies both criteria.

[Reason]    First, I need to break it down.
    The main points are: (1) Bill
Woodfull was a cricketer (Australian Test
    captain, probably early 20th century);
(2) the question is about who he opened the
    batting with (i.e., his opening
partner in cricket); (3) that partner must
    have died in April 1991.
So the steps would be: first, find out who
    Bill Woodfull's opening partners
were during his career. Then, check which
    of those players died in April 1991.
Starting with the first part: identifying
    Bill Woodfull's opening partners.
Let me do a search for that.

[Search]    Bill Woodfull opening partner
    cricket
[Result]    Bill Woodfull's most
    significant and regular opening
    partnership was
with Bill Ponsford, described as a ``fellow
    Victorian'' and part
of a historic opening stand; other partners
    mentioned are Warren Bardsley and
Charlie Macartney. No death dates are
    provided.
[Guide]     Insufficient (confidence 100%).
    Feel free to explore alternative
            paths, such as trying a
    different search query or taking other
            retrieval steps as needed, to
    derive an answer to the question [...].
            Please think carefully, and use
    up to 230 words.

[Reason]    Ponsford is the most prominent
    candidate. Following step (2), verify
his death date through retrieval rather
    than relying on memory.

[Search]    Bill Ponsford death date
[Result]    Two documents confirm: "Bill
    Ponsford (19 October 1900– 6 April 1991) was
    an Australian
    cricketer"; ``He died there on 6 April
1991; at the time he was Australia's oldest
    living Test cricketer.''
[Guide]     Sufficient. 1) Derive an
    intermediate answer and continue toward
            the next retrieval if more is
    needed. 2) Otherwise, provide the
            final answer in the format
    \boxed{YOUR_ANSWER}.

[Reason]    Both criteria are satisfied
    directly from retrieval: Ponsford was
Woodfull's opening partner and died on 6
    April 1991.

[Answer]    Bill Ponsford   (correct, F1 =
    1.00)
```

Trace annotation. RE-GUIDE preserves the initially retrieved candidate and converts the unresolved second hop into a targeted verification query. The Insufficient verdict on the first retrieval reflects that the agent's preceding reasoning had articulated both sub-tasks—partner identification and April-1991 death-date verification—as the information need, of which only partner identification was resolved by the retrieved evidence (which explicitly notes "No death dates are provided"). Instead of expanding the candidate set, RE-GUIDE checks the missing death-date constraint directly and terminates once both conditions are supported by retrieval.

Observation. Both agents identify Bill Ponsford as the correct opening-partner candidate in the first retrieval, but they diverge at the second hop, where the death-date constraint must be verified. SEARCH-O1 replaces this verification step with an unsupported memory-based assumption, incorrectly rules out Ponsford, and then spends the remaining retrieval budget searching for alternative candidates. As a result, the trajectory expands into a long but unproductive search process and reaches the 20-call cap without producing an answer.

In contrast, RE-GUIDE turns the unresolved second hop into a targeted retrieval query, "Bill Ponsford death date," which directly returns evidence that Ponsford died on 6 April 1991. This case shows that RE-GUIDE mitigates over-thinking not merely by shortening trajectories, but by redirecting the agent from speculative candidate expansion to evidence-grounded verification.