RationalSearch: Need-Aware Search Triggering with Reinforcement Learning

Anonymous ACL submission

Abstract

Recently, search-augmented reinforcement learning (RL) agents have demonstrated strong knowledge-reasoning capabilities by leveraging external information. However, existing RL-based agents often resort to search as a shortcut to correctness, rather than learning *when* to take tool actions in response to information deficits. Empirically, tool calls keep rising while accuracy gains saturate, exposing the need-use misalignment. Since optimization is largely outcome-only, necessity-aligned and shortcut-triggering search behaviors are not separable under such objectives, leaving the *when*-to-search decisions unconstrained. To make necessity-aligned search behavior verifiable, this paper proposes the rational search score (RSS), a need-use alignment metric comparable across model capability and task difficulty. Concretely, RSS combines two signals for behavioral diagnosis: a high-diversity no-search evaluation to estimate deficit, and a task-conditioned band to assess tool use. To validate RSS as an effective objective for optimizing when-to-search behavior, we use it to fine-tune Qwen2.5-3B/7B-Instruct into Rational-R1. Main experiments show that RSS-guided optimization makes necessity-aligned search triggering learnable, improving the accuracy–cost trade-off while maintaining strong accuracy with better-aligned tool use. Using RSS for behavioral diagnosis, we further find that current RL-based baselines learn *how* to use search, but not *when* to invoke it. [1]

1 Introduction

Large-scale reinforcement learning with verifiable rewards has emerged as a powerful paradigm for improving large language models (LLMs), yielding strong performance on reasoning benchmarks by encouraging "deep-thinking" reasoning traces (Liu et al., 2024; Jaech et al., 2024; Team et al., 2025; Li et al., 2025d; He et al., 2024; Cobbe et al., 2021; Hendrycks et al., 2021). However, real-world reasoning tasks such as coding (Zhuo et al., 2025), knowledge-intensive QA (Mallen et al., 2023; Aksitov et al., 2023), and deep search (Phan et al., 2025; Hu et al., 2025) often require external information (Guo et al., 2024). This motivates tool-augmented RL (Dong et al., 2025c,a; Shang et al., 2025; Xue et al., 2025), where LLMs must decide *when* and *how* to interact with the environment and integrate tool outputs into reasoning.

Current search-augmented RL models (Mei et al., 2025; Zhao et al., 2025a) like Search-R1 (Jin et al., 2025a) increasingly rely on search as a shortcut to reduce factual errors and improve accuracy. However, empirical results show that accuracy often saturates as tool calls increase (Fig. 1a) (Li et al., 2025c; Prato et al., 2024). In particular, search can be triggered even when the needed information is already recoverable from the model's parametric knowledge, leading to a high over-query rate (OOR) (Fig 1b, left). Together, these observations suggest that current models mainly learn *how* to use search, but not *when* to use it (Fig 1, right). Meanwhile, existing methods (Damani et al., 2025; Wang et al., 2025b; Moskvoretskii et al., 2025) lack a principled, comparable metric for rational search triggering across models and datasets. Neither behavioral usage metrics nor model-probing signals (e.g., confidence or ECE (Pavlovic, 2025)) solve this problem, because both conflate information need with capability and difficulty (Fig. 1b).

To diagnose and optimize search-triggering behavior, this paper proposes the rational search score (RSS), which characterizes rationality along two dimensions: information need and tool use. RSS estimates model-specific information need via high-diversity no-search evaluation and assesses whether tool use falls within a task-conditioned range. As shown in Fig. 1c (left), RSS deems search rational when tool use matches estimated information need.

[1] The code is https://anonymous.4open.science/Rational-R1.

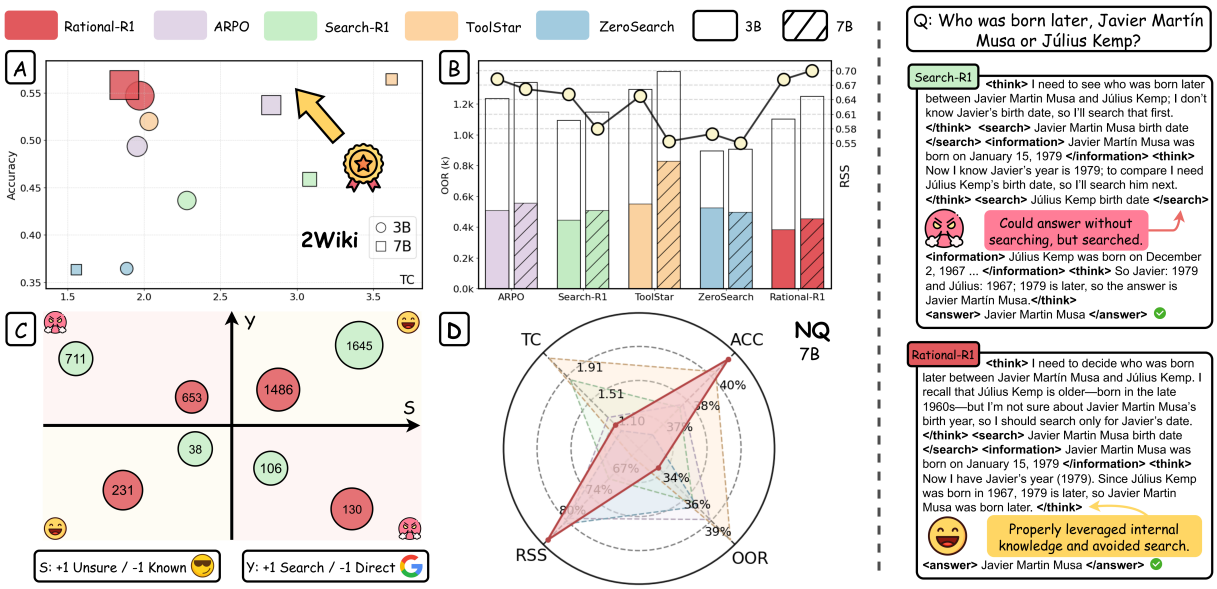

Figure 1: RSS motivates and diagnoses rational search. (A) Accuracy vs. tool calls (TC) on a 2WikiMultiHop-2500 subset. (B) On the same subset, bars show over-query rate (OOR; left axis) and the line shows RSS (right axis) for each method/model size. (C) RSS quadrant breakdown on 2Wiki (3B): Rational-R1 vs. Search-R1. (D) Summary on NQ comparing TC, EM, RSS, and OOR. Right panel: A representative rational-search example comparing Search-R1 and Rational-R1. Baselines include representative RL-based search agents.

Thus, RSS defines rational search not as minimal search, but as tool use aligned with information need. Importantly, RSS also provides a training signal for learning *when* to search by encouraging need-aware tool use and discouraging mismatches. Building on RSS, we fine-tune 3B/7B agents into Rational-R1, showing that they can learn not only *how* to use search tools but also *when* to invoke them (Fig. 1d, left).

Rational-R1 achieves strong accuracy on seven knowledge-intensive benchmarks while reducing unnecessary tool calls. Compared with accuracy-only RL, RSS-guided training improves *when*-to-search behavior rather than merely relying on more search. Beyond optimization, RSS also serves as an interpretable behavioral diagnostic, clearly separating need-aware from mismatched tool use across model sizes and difficulty levels. Empirical analyses further support both RSS design choices: diverse no-search sampling sharpens the capability-gap proxy, while task-conditioned call ranges distinguish under-querying from over-querying.

2 Related Work

Reinforcement Learning for Search-Augmented LLM Agents. Recent progress in RL post-training has improved LLM reasoning ability (Xi et al., 2025; Aggarwal and Welleck, 2025; Wang et al., 2026). Motivated by this progress, early work (Feng et al., 2025a; Chen et al., 2025d,c; Feng et al., 2025b) enables models to interact with search engines and incorporate external evidence into reasoning. Representative methods differ mainly in how search is integrated and optimized: (1) Search-R1 (Jin et al., 2025a) integrates multi-turn retrieval into RL rollouts via token-level triggers; (2) ZeroSearch (Sun et al., 2025a) replaces live search with an SFT-trained LLM retriever simulator under curriculum RL; (3) ToolStar (Dong et al., 2025b) bootstraps multi-tool use from synthesized trajectories and optimizes it with self-critic RL under hierarchical rewards; and (4) ARPO (Dong et al., 2025c) performs branch-level credit assignment after high-entropy post-tool steps. We defer knowledge-boundary and search-regulation discussion to Appendix A.

3 System Formulation and Optimization for Rational Search Agents

3.1 Preliminary

Environment. An environment (Wang et al., 2025a) is a tuple $E = (W, \mathrm{Exec})$, where the world information store decomposes as $W = W_{\mathrm{int}} \cup W_{\mathrm{out}}$. Here, W_{int} denotes the model's internal knowledge, while W_{out} denotes external information accessible through tool outputs. The execution semantics $\mathrm{Exec} = (\mathcal{T}_{\mathrm{int}}, \mathcal{T}_{\mathrm{out}}, \mathrm{Upd})$ specify inter-

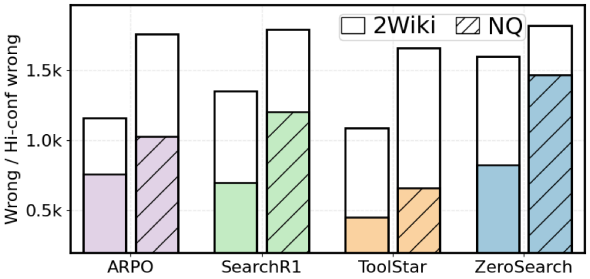

Figure 2: Confident-but-wrong errors under protocol-preserving no-search probing on 2Wiki and NQ with 7B agents. The outer contour denotes all incorrect predictions, and the colored inner region marks high-confidence errors. Their persistence shows that confidence can reflect answer commitment rather than internal sufficiency, motivating outcome-based deficit estimation via the no-search correctness rate p_i.

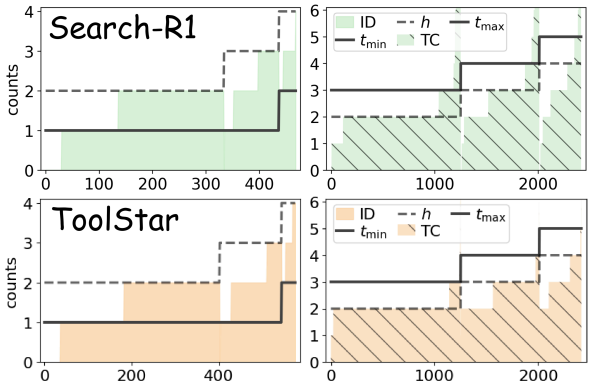

Figure 3: Sanity check for the hop-conditioned adequacy band on MuSiQue with 3B agents. Using gold sub-question count as the structure cue h, the left panel compares empirical information-deficit count (ID) with h, while the right panel checks realized tool calls against $[t_{\min}(h), t_{\max}(h)]$. Successful trajectories broadly fall within the band, supporting it as a verifiable adequacy filter rather than a fixed oracle budget.

nal tools, external tools, and how returned information is merged into the information state.

Agent system. We study the single-model setting, where an agent is implemented by $m = (\pi_m, W_{\text{int}}(m))$. The policy π_m selects typed actions conditioned on the current state, and $W_{\text{int}}(m)$ contributes the model-specific internal knowledge. The resulting agent system is

$$\Sigma(m) = \big((W_{\text{out}} \cup W_{\text{int}}(m), \text{Exec}), \{\pi_m\}\big). \quad (1)$$

A rational search agent should use π_m to access W_{out} only when $W_{\text{int}}(m)$ is insufficient, rather than treating external search as a generic shortcut.

Interaction process. Tool-augmented reasoning iteratively updates an information state $s_t = (o_t, I_t)$ through internal and external actions (Yao et al., 2023). At each step, π_m selects $a_t \in \{\texttt{int}(\cdot), \texttt{out}(\cdot), \texttt{stop}(\cdot)\}$, where $\texttt{int}(\cdot)$ invokes internal reasoning, $\texttt{out}(\cdot)$ invokes external information, and $\texttt{stop}(\cdot)$ returns the final output y. Let Z_{int} and Z_{out} denote the internal and external information collected during an episode, respectively, and let $c_i = |Z_{\text{out}}|$ be the number of external calls. Thus, tool-augmented RL must learn not only *how* to use external information, but also *when* it is needed, which calls for a capability-aware alignment between information need and tool use.

3.2 Rational Search Score for Need-Use Alignment

Motivated by need-use mismatch in search-augmented agent systems, we introduce the rational search score (RSS), which combines an information-deficit signal with a task-conditioned tool-use signal to quantify how well a policy π_m aligns external search actions $\texttt{out}(\cdot)$ with its information need. Appendix B further validates RSS through high-diversity no-search analyses, robustness of deficit choices, adequacy-band analysis, and empirical robustness across RSS variants.

Information-deficit signal. The information-deficit signal indicates whether external access to W_{out} (via $\texttt{out}(\cdot)$) is required because $W_{\text{int}}(m)$ together with internal computation cannot supply the missing information needed for a correct terminal output. Since $W_{\text{int}}(m)$ is a black box, directly assessing this deficit at intermediate states is infeasible. We therefore operationalize the deficit at the episode level by running the policy with external actions disabled (i.e., $Z_{\text{out}} = \emptyset$). Concretely, we keep the tool-use template unchanged and replace each retrieval return with the same fixed no-evidence stub, `"This experiment will NOT return retrieval results; rely solely on your internal knowledge and reasoning."`, so that prompts and interaction format are preserved while the external evidence channel carries no task-specific information. For each input x_i with ground-truth answer g_i, we draw K no-search outputs $\{\hat{y}_i^{(k)}\}_{k=1}^{K} \sim \pi_m(\cdot \mid x_i)$ using high-diversity decoding, and compute

$$p_i = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\left[\text{Corr}\left(\hat{y}_i^{(k)}, g_i\right) = 1\right]. \quad (2)$$

We define the deficit indicator as $S_i = \mathbf{1}[p_i < \tau]$. This outcome-based estimate is preferred over

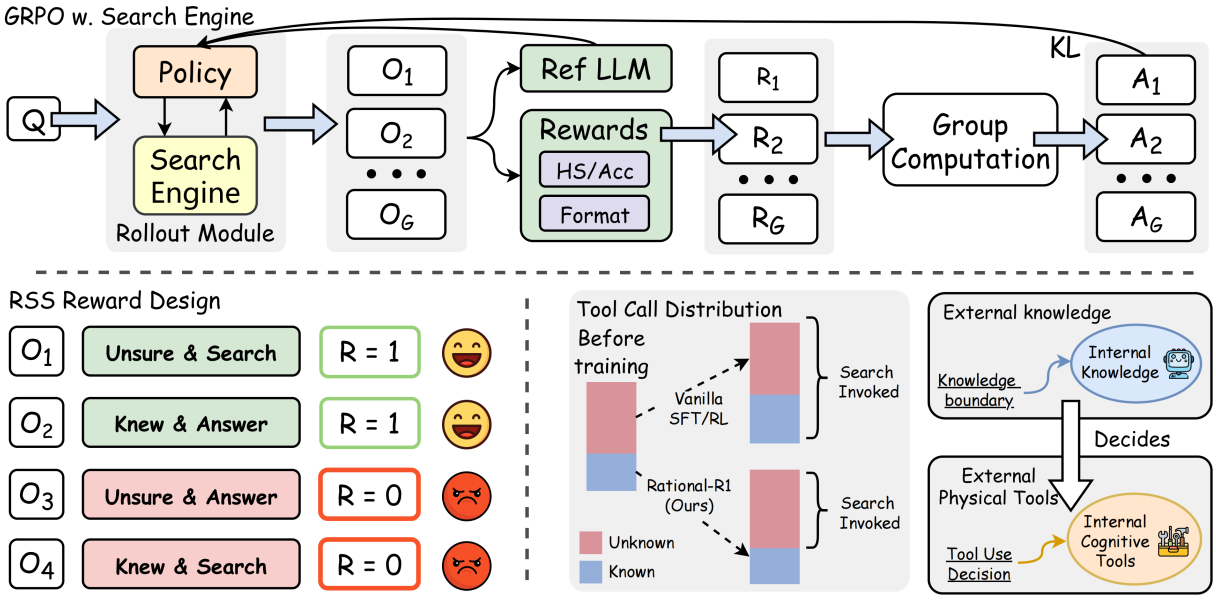

Figure 4: RSS-guided GRPO for rational search triggering. Under $\Sigma(m)$, π_m decides whether to invoke $\texttt{out}(\cdot)$ to access W_{out} or rely on $W_{\text{int}}(m)$. The terminal reward combines answer correctness, format, and RSS, which rewards adequate search under deficit and limited search otherwise. Practical overhead analysis for RSS-guided training is deferred to Appendix C.

confidence-only probing, since high-confidence no-search predictions can still be wrong (Fig. 2).

Task-conditioned tool-use signal. Given the deficit label S_i, we summarize the agent's realized external interaction at the episode level. Let $c_i \triangleq \sum_{t<T} \mathbf{1}[a_t = \texttt{out}(\cdot)] = |Z_{\text{out}}|$ be the number of external tool calls, and let h_i denote a task-structure proxy. We evaluate tool use against a hop-conditioned tolerance band via

$$Y_i \triangleq \mathbf{1}[t_{\min}(h_i) \le c_i \le t_{\max}(h_i)]\,, \qquad (3)$$

where $t_{\min}$ and $t_{\max}$ are monotone in h_i. This range-based check refines the coarse deficit proxy without requiring state-by-state deficit evaluation. Fig. 3 validates this band as a verifiable adequacy filter rather than an oracle budget.

Rational Search Score. Given the deficit indicator S_i and the task-conditioned use indicator Y_i, we define the RSS for episode i as

$$\text{RSS}_i \triangleq S_i \cdot Y_i \;+\; (1 - S_i) \cdot \mathbf{1}[c_i \le t_0], \qquad (4)$$

and the dataset-level RSS as

$$\text{RSS} \triangleq \frac{1}{N} \sum_{i=1}^{N} \text{RSS}_i, \qquad (5)$$

where t_0 is a small slack for the no-deficit case (default $t_0 = 1$). In RSS, S_i provides a necessity signal for tool use, while Y_i specifies a task-aware notion of adequate interaction; the slack for the no-deficit case prevents unnecessary repeated search when no deficit is estimated. Together, they define a capability-aware and comparable notion of rational search triggering: a policy π_m should invoke $\texttt{out}(\cdot)$ to access W_{out} only when $W_{\text{int}}(m)$ is insufficient. Because S_i is model-specific and Y_i is task-conditioned, RSS remains comparable across model sizes and task difficulties.

3.3 RSS-Guided RL for When-to-Search

Under the interaction system $\Sigma(m)$, the policy π_m repeatedly decides whether to rely on internal capability $W_{\text{int}}(m)$ via $\texttt{int}(\cdot)$ or to trigger $\texttt{out}(\cdot)$ to access external evidence from W_{out}. RSS makes need-use alignment a verifiable terminal signal, so RL fine-tuning directly targets *when* to search. Following search-augmented RL setups (Jin et al., 2025a), we optimize a policy $\pi_\theta(\cdot \mid x; R)$ that interleaves generation with calls to a search engine R. Given $x \sim \mathcal{D}$, we sample an interleaved rollout $y \sim \pi_{\text{old}}(\cdot \mid x; R)$ containing both model-generated tokens and retrieved spans inserted into the context. We consider the standard KL-regularized objective:

$$\max_{\theta} \quad \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\text{old}}(\cdot|x;R)}\Big[r_\phi(x,y)\Big] \\ - \beta\, D_{\text{KL}}\Big(\pi_\theta(\cdot \mid x; R) \,\big\|\, \pi_{\text{ref}}(\cdot \mid x; R)\Big), \qquad (6)$$

where π_{ref} is a frozen reference policy and r_ϕ is a terminal reward. Since y interleaves generated tokens with retrieved evidence, we mask all retrieved spans when computing both the policy-gradient loss and the KL term, and update only LLM-generated tokens. This prevents the policy from gaining gradient signal by trivially copying tool outputs and isolates learning on decision and reasoning tokens. As illustrated in Fig. 4, we instantiate this objective with RSS-guided GRPO, so that the model learns not only *when* to trigger $\texttt{out}(\cdot)$ according to its internal information need, but also *how* to use the retrieved evidence effectively once external search is invoked.

Starting from Qwen2.5-3B/7B-Instruct (Qwen et al., 2025), we obtain Rational-R1 (3B/7B) by fine-tuning with RSS-guided GRPO. For stability and need-aware curriculum, we construct RATIONALSEARCHMIX with a lightweight SFT cold start, followed by Stage-1 RL examples with lower hop h for basic triggering behavior and Stage-2 RL examples with higher h and longer-horizon search interactions. We compare the curriculum schedule (Stage1→Stage2) with a mixed-all schedule on the same validation set for controlled analysis; detailed training setup is deferred to Appendix C.

4 Main Experiments

4.1 Experiments Setup

Datasets. We evaluate Rational-R1 on 7 knowledge-intensive benchmarks, categorized as follows: (1) **General Question Answering**: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); (2) **Multi-Hop Question Answering**: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023).

Baselines. To evaluate Rational-R1, we compare against three groups of baselines: (1) **Inference without Retrieval**: Direct inference and Chain-of-Thought (CoT) (Wei et al., 2023); (2) **Inference with Retrieval**: Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) and IRCoT (Trivedi et al., 2023); (3) **RL-Based Fine-Tuning Methods**: Search-R1 (Jin et al., 2025a), ARPO (Dong et al., 2025c), ToolStar (Dong et al., 2025b), and ZeroSearch (Sun et al., 2025a). We also include OTC (Wang et al., 2025b) as a contextual low-call reference, since it mainly emphasizes tool-cost reduction rather than need-use alignment. For the SFT→RL diagnosis in Sec. 5, we reproduce baseline SFT checkpoints under our unified execution setup. Details are in Appendix C.5.

Experimental Setup. Experiments are conducted with Qwen-2.5-3B/7B-Instruct (Qwen et al., 2025). For Wikipedia-based retrieval, we use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2024) as the retriever, retrieving top-3 passages per query for all retrieval-based methods following Search-R1 (Jin et al., 2025a; Lin et al., 2023) for fair comparison. Accuracy is measured by exact match (Yu et al., 2024), and tool cost by the number of external calls $\text{TC} = c_i = |Z_{\text{out}}|$. Beyond the 2Wiki/NQ analyses in Fig. 1, we report RSS diagnostics on PopQA, MuSiQue, and Bamboogle.

Implementation Details. RSS-guided training is implemented with Sky-RL (Cao et al., 2025), excluding tool-call results from the loss to avoid bias. Unless otherwise specified, training follows Search-R1 for fair comparison, and uses the same Wikipedia retriever as in evaluation. For RSS, we set $\tau = 0.5$ and estimate no-search correctness with $K = 4$. We further use a hop-conditioned adequate range $[t_{\min}(h),\, t_{\max}(h)]$, where $t_{\min}(h) = \max(1, \lfloor h/2 \rfloor)$ and $t_{\max}(h) = h+1$; for no-deficit cases, we allow a slack $t_0 = 1$. Additional training and analysis details are provided in Appendix D.

4.2 Main Results

The main results comparing Rational-R1 with baseline methods across the seven datasets are presented in Table 1. From the results, we make the following key observations: (1) **Rational-R1 consistently outperforms strong baseline methods.** These gains hold across both in-domain and out-of-domain evaluation. (2) **Scaling improves *how* to search but not automatically *when* to search.** From 3B to 7B, tool-augmented agents are generally more accurate yet also make more external calls c_i. This pattern suggests that stronger internal capability does not automatically calibrate external search: additional $\texttt{out}(\cdot)$ calls may return fragments with limited marginal value beyond what is already recoverable internally. (3) **Neither lower TC nor higher accuracy implies rational search.** Both metrics conflate information need, model capability, and task difficulty, and thus cannot diagnose whether $\texttt{out}(\cdot)$ is triggered only to fill the remaining gaps beyond $W_{\text{int}}(m)$. Stronger models may also induce more $\texttt{out}(\cdot)$ calls through longer reasoning chains, even when the added Z_{out} is in-

Methods	General QA			Multi-Hop QA				Avg	
	TriviaQA†	NQ†	PopQA⋆	HotPot†	2Wiki†	MuSiQue⋆	Bam⋆	Acc (↑)	TC
Qwen2.5-3B-Instruct									
Direct Inference	0.288	0.106	0.108	0.149	0.244	0.020	0.024	0.134	–
CoT	0.032	0.023	0.005	0.021	0.021	0.002	0.000	0.015	–
IRCoT	0.312	0.111	0.200	0.164	0.171	0.067	0.240	0.181	–
RAG	0.544	0.348	0.387	0.255	0.226	0.047	0.080	0.270	–
OTC-GRPO⋄	0.608	0.444	0.441	0.365	0.341	0.124	0.266	0.366	1.008
Search-R1	0.630	0.341	0.378	0.388	0.319	0.160	0.264	0.354	1.946
ZeroSearch	0.638	0.338	0.452	0.356	0.364	0.102	0.328	**0.368**	**1.467**
ToolStar	0.711	**0.361**	0.468	0.476	0.520	0.202	0.520	0.465	1.833
ARPO	0.699	0.356	0.490	0.440	0.494	0.187	0.456	0.446	1.565
Rational-R1-3B	**0.745**	0.357	**0.509**	**0.487**	**0.547**	**0.217**	**0.536**	**0.485**	1.549
Qwen2.5-7B-Instruct									
Direct Inference	0.408	0.134	0.140	0.183	0.250	0.031	0.120	0.181	–
CoT	0.185	0.048	0.054	0.092	0.111	0.022	0.232	0.106	–
IRCoT	0.478	0.224	0.301	0.133	0.149	0.072	0.224	0.226	–
RAG	0.585	0.349	0.392	0.299	0.235	0.058	0.208	0.304	–
OTC-GRPO⋄	0.597	0.444	0.431	0.366	0.311	0.130	0.250	0.361	0.990
Search-R1	0.745	0.393	0.397	0.432	0.414	0.198	0.368	0.421	2.332
ZeroSearch	0.607	0.359	0.511	0.380	0.363	0.161	0.296	0.382	**1.625**
ToolStar	0.738	0.405	**0.516**	0.520	**0.564**	0.261	**0.568**	0.510	2.875
ARPO	0.724	0.381	0.471	0.493	0.537	0.231	0.504	0.477	1.809
Rational-R1-7B	**0.783**	**0.414**	0.513	**0.544**	0.558	**0.279**	0.550	**0.520**	1.653

Table 1: Main results under the standard accuracy–cost view. **Bold** marks the best accuracy within each model size. TC is a cost statistic, not a rationality criterion: lower TC may indicate either calibrated abstention or under-searching. †/⋆ indicate in-/out-of-domain evaluation. ⋄ denotes OTC-GRPO as a contextual low-call reference.

creasingly redundant. Thus, lower TC does not imply better calibration, and higher accuracy does not imply more rational triggering. For example, ZeroSearch and OTC-GRPO reduce TC at the cost of lower accuracy, while ToolStar attains strong accuracy with substantially higher TC. As a result, Rational-R1 validates RSS as a rational-behavioral supervision signal for *when*-to-search across model sizes by calibrating π_m to trigger $\texttt{out}(\cdot)$ mainly when $W_{\text{int}}(m)$ does not suffice. Table 2 complements these results with need-use alignment, beyond raw tool cost.

Methods	PopQA		Bam		MuSiQue		Avg.
	TC	**RSS**	**TC**	**RSS**	**TC**	**RSS**	**Acc**
3B							
Search-R1	1.646	0.862	2.240	0.744	2.703	0.943	0.336
ZeroSearch	1.349	0.842	**1.632**	0.488	**1.759**	0.875	0.294
ToolStar	1.410	0.843	1.984	0.696	2.428	0.953	0.397
ARPO	1.168	0.836	1.840	0.696	2.041	0.931	0.378
Rational-R1	**1.074**	**0.884**	1.792	**0.776**	2.925	**0.967**	**0.421**
7B							
Search-R1	1.961	0.809	2.336	0.672	3.081	0.923	0.363
ZeroSearch	1.035	0.829	1.032	0.412	1.989	0.670	0.323
ToolStar	2.409	0.810	3.024	0.568	3.686	0.903	**0.448**
ARPO	1.265	0.815	1.832	0.568	2.307	0.887	0.402
Rational-R1	1.263	0.846	1.888	**0.760**	2.729	**0.924**	0.447

Table 2: RSS-based behavioral diagnosis across benchmarks and model scales. For each dataset, we report average TC and RSS; the last column reports average accuracy over PopQA, Bam, and MuSiQue. Additional RSS analyses on 2Wiki and NQ are provided in Fig. 1.

5 Analysis

Unless otherwise noted, analyses focus on 3B; 7B and deep-search results are in Appendix E.

Behavioral Diagnosis with RSS Across Benchmarks. Table 2 reports TC and RSS across three benchmarks, together with average accuracy, for both 3B and 7B agents. We highlight three observations: (1) RSS is more reflective of need-use alignment than raw tool-call count across benchmarks with different difficulty profiles. Across 3B baselines, TC rises substantially from PopQA (1.17–1.65) to MuSiQue (2.04–2.70) and remains high on Bam (1.63–2.24), yet RSS does not follow

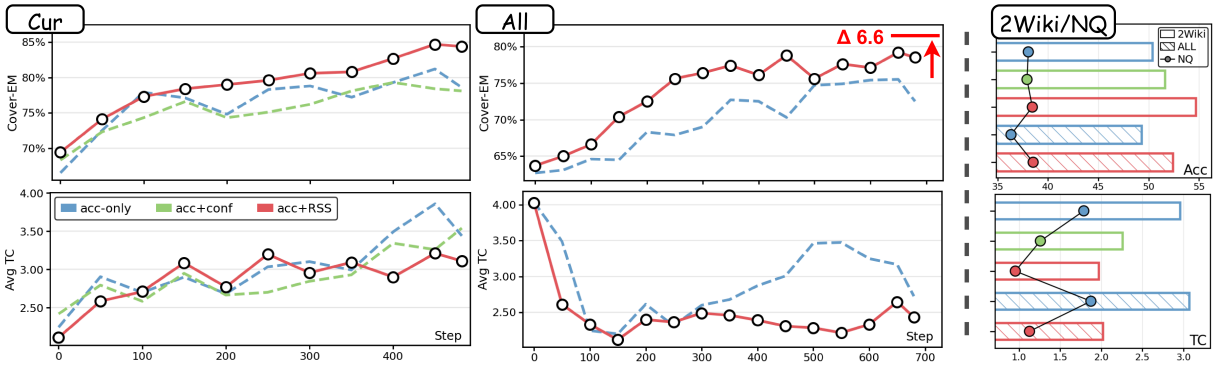

Figure 5: Training dynamics under different GRPO objectives (`acc-only`, `acc+conf`, `acc+RSS`) and training schedules (Cur: Stage1→Stage2; All: mixed), reporting cover-EM (top) and average tool calls (bottom) over steps.

Method	Easy		Hard	
	Acc	TC	Acc	TC
PopQA				
Search-R1-3B	72.81	1.36	36.77	1.73
Rational-R1-3B	**89.47**	1.10	48.19	1.40
Search-R1-7B	75.55	1.60	36.65	2.09
Rational-R1-7B	78.09	**0.93**	**50.90**	**1.30**
Bamboogle				
Search-R1-3B	67.92	2.28	30.56	2.21
Rational-R1-3B	72.22	1.85	36.58	2.42
Search-R1-7B	69.23	2.17	32.00	2.52
Rational-R1-7B	**84.78**	**1.59**	**40.97**	**1.89**

Table 3: Easy/hard calibration view using same-backbone no-search pass@4. Easy: pass@4 > 0; hard: pass@4 $= 0$.

a monotonic trend, suggesting that RSS captures need-use alignment rather than difficulty-induced call frequency. (2) Scaling capability does not guarantee higher RSS. When scaling from 3B to 7B, TC often increases, but RSS does not consistently improve: on PopQA, Search-R1 increases TC (1.646→1.961) while RSS drops (0.862→0.809); moreover, at 7B ToolStar uses far more calls than Search-R1 (2.409 vs. 1.961) yet attains essentially the same RSS (0.810 vs. 0.809), suggesting that neither higher capability nor higher cost implies more need-aligned triggering. (3) RSS distinguishes necessary cost from degenerate cost-cutting within a dataset. On Bam, fewer calls can coincide with lower RSS, indicating under-evidence rather than rational abstention. On MuSiQue, higher TC can still yield the highest RSS, indicating that additional calls are concentrated on deficit cases. Taken together, RSS provides a stable behavioral axis for diagnosing need-use alignment.

A Direct Calibration View via Easy/Hard Buckets. To provide a more direct instance-level view of search calibration, we bucket questions by same-backbone no-search pass@4: questions with pass@4 > 0 are treated as easy, and those with pass@4 $= 0$ as hard. This split uses same-backbone no-search solvability as a simple calibration-oriented proxy, without relying on RSS itself. Table 3 shows that Rational-R1 reduces tool calls on easy questions while improving accuracy on hard questions, which is more consistent with necessity-aware triggering than indiscriminate cost cutting. This pattern suggests that Rational-R1 searches less when internal knowledge suffices, while preserving or using search more effectively when additional evidence is needed.

Training Objectives Guide When-to-Search. We fine-tune Rational-R1 with GRPO and vary two factors: the reward objective (`acc-only`, `acc+conf`, `acc+RSS`) and the training schedule (Cur: Stage1→Stage2 vs. All: mixed), as shown in Fig. 5. We highlight three observations. (1) RSS sustains late-stage accuracy gains. Under both Cur and All, `acc-only` improves quickly but plateaus, whereas `acc+RSS` maintains a clear late-stage advantage. By making *when* to search verifiable through $t_{\min}(h) \leq c_i \leq t_{\max}(h)$, RSS rewards adequate evidence acquisition on harder (S_i=1) instances, creating better conditions for using retrieved evidence effectively. (2) RSS yields more stable tool usage. In the TC curves, `acc+RSS` is consistently more stable, whereas `acc-only` exhibits larger drift. In contrast, `acc+conf` can reduce TC without improving accuracy, consistent with the non-trivial confident-but-wrong cases under no-search probing. (3) Curriculum improves learnability and stability. Compared with All, Cur reaches strong accuracy earlier (top) and shows fewer TC drifts (bottom), indicating that learning

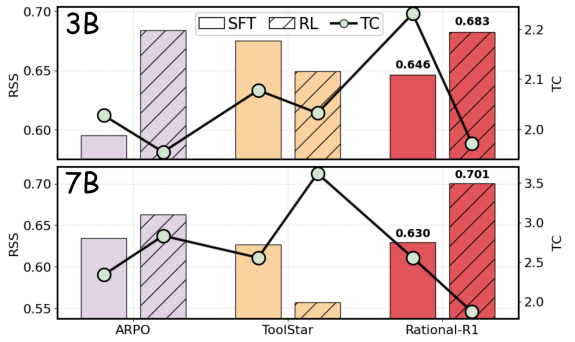

Figure 6: SFT vs. RL on 2WikiMultiHop. Bars show RSS and the line shows average TC for SFT and RL checkpoints of representative search-RL agents.

Method	(+,+)	(+,-)	(-,-)	(-,+)	TC
ARPO					
SFT	1475	125	4	896	2.03
RL	1710	107	2	755	1.95
ToolStar					
SFT	1682	62	6	750	2.08
RL	1623	68	0	809	2.03
Rational-R1					
SFT	1606	112	10	772	2.23
RL-ACC	1641	109	0	750	2.12
RL-RSS	1486	130	231	653	1.97

Table 4: Quadrant breakdown on 2WikiMultiHop (3B). The first sign denotes information deficit S and the second denotes tool-use adequacy Y: $(+,+)$ need-aligned search, $(+,-)$ under-evidence, $(-,-)$ need-aligned abstention, and $(-,+)$ over-triggering.

a stable tool-use behavior first makes it easier to scale to harder stages. Finally, Fig. 5 (right) shows that acc+RSS achieves stronger accuracy with more stable TC at convergence on both 2Wiki and NQ, supporting RSS as a better objective for learning *when* to trigger $\texttt{out}(\cdot)$.

Why RL Does Not Learn When to Search from Accuracy Alone. To isolate the effect of RL fine-tuning, Fig. 6 compares SFT and RL checkpoints of representative search-RL agents (ARPO, ToolStar, Rational-R1) at 3B and 7B in terms of RSS and TC. For fair comparison, we reproduce the SFT baselines under our unified pipeline (Appendix C.5). SFT→RL can improve the tool-action competence of π_m, but this does not reliably translate into more calibrated triggering: TC (c_i) often fails to decrease or stabilize. The reason is that accuracy-driven RL rewards correct outcomes and successful tool use, yet leaves need-use mismatch ($S_i \neq Y_i$) largely unconstrained, so c_i has little pressure to converge to a need-aligned range. The gap is clearer at 7B: stronger $W_{\text{int}}(m)$ can improve *how* to use $\texttt{out}(\cdot)$ while RSS drops and TC increases for some baselines, showing that scaling alone does not fix *when* to search. Therefore, RSS-guided RL is needed to train π_m to better align $\texttt{out}(\cdot)$ triggers with estimated insufficiency of $W_{\text{int}}(m)$.

What Goes Wrong: A Quadrant-Level Diagnosis of Search Triggering. The previous analysis suggests an objective gap: optimizing end accuracy alone does not reliably align tool use with information need. To trace this gap to a concrete behavioral source, Table 4 breaks down RSS quadrants on the 2WikiMultiHop dev subset for 3B agents, comparing ToolStar (SFT vs. RL) and Rational-R1 (SFT, RL-ACC, RL-RSS). This decomposition explains Fig. 1a/b: for baseline agents, most extra tool calls come from unnecessary triggers in $(-,+)$ rather than missed searches under $(+,-)$, so TC keeps growing while accuracy gains quickly saturate. For ToolStar, SFT→RL does not systematically curb over-triggering: the dominant mass remains in $(-,+)$, while $(-,-)$ stays near-empty, leaving little pressure for TC to settle into a need-aligned range. Our controlled ablation isolates the missing signal: RL-ACC improves *how* to use search but leaves unnecessary triggers largely intact, whereas RL-RSS makes *when* to search verifiable and shifts behavior from $(-,+)$ toward need-aligned abstention $(-,-)$, reducing redundant calls. Together with Fig. 6, the breakdown links accuracy saturation to cost-heavy over-triggering behavior, and shows that RSS supplies the missing signal to suppress redundant triggering.

6 Conclusion

Overall, this paper introduces the rational search score (RSS), a capability-aware metric that makes *when*-to-search behavior comparable and optimizable. Our analyses show that strong search-RL baselines mainly improve *how* to use search, while *when* to invoke it remains weakly calibrated, with redundant triggering emerging as a major source of accuracy saturation. Building on RSS, we train Rational-R1, enabling agents to learn not only *how* to use external search effectively, but also *when* external evidence is actually needed. These results suggest that rational search is determined not by tool-use volume alone, but by whether search is aligned with estimated insufficiency of internal knowledge and the need for external evidence.

Limitations

Despite strong empirical results on knowledge reasoning tasks, several limitations remain.

- **Scope of RSS.** RSS is designed as a verifiable signal for *when* to trigger external information seeking in search-augmented reasoning, and we do not claim it as a universal metric for arbitrary tool types, especially execution-centric tools such as code/math solvers or verifiers.
- **Task-structure proxy.** The current adequacy check instantiates task structure using a coarse proxy, such as hop count, to keep the criterion verifiable and comparable; tasks without explicit structure annotations may require alternative proxies.
- **Evaluation scope.** Our controlled comparisons are conducted mainly within the Qwen2.5 3B/7B family under a unified search protocol, so transfer across architectures, retrievers, and tool protocols requires further validation.

Future work includes developing finer-grained evaluation beyond band-based adequacy checks and extending validation to frontier-scale deep-search settings and closed-source agents.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, Manzil Zaheer, Felix Yu, and Sanjiv Kumar. 2023. Rest meets react: Self-improvement for multi-step reasoning llm agent. *Preprint*, arXiv:2312.10003.

Shiyi Cao, Dacheng Li, Fangzhou Zhao, Shuo Yuan, Sumanth R. Hegde, Connor Chen, Charlie Ruan, Tyler Griggs, Shu Liu, Eric Tang, Richard Liaw, Philipp Moritz, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Skyrl-agent: Efficient rl training for multi-turn llm agent. *Preprint*, arXiv:2511.16108.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025a. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *Preprint*, arXiv:2506.13651.

Lida Chen, Zujie Liang, Xintao Wang, Jiaqing Liang, Yanghua Xiao, Feng Wei, Jinglei Chen, Zhenghong Hao, Bing Han, and Wei Wang. 2025b. Teaching large language models to express knowledge boundary from their own signals. In *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, pages 26–39, Vienna, Austria. Association for Computational Linguistics.

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

Zhuo Chen, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinyu Geng, Pengjun Xie, Fei Huang, and Kewei Tu. 2025d. Detecting knowledge boundary of vision large language models by sampling-based inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28677–28692, Suzhou, China. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *Preprint*, arXiv:2507.16806.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025a. Agentic entropy-balanced policy optimization. *Preprint*, arXiv:2510.14545.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2025b. Toolstar: Empowering llm-brained multi-tool reasoner via reinforcement learning. *Preprint*, arXiv:2505.16410.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025c. Agentic reinforced policy optimization. *Preprint*, arXiv:2507.19849.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025a. Retool: Reinforcement learning for strategic tool use in llms. *Preprint*, arXiv:2504.11536.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Guochao Jiang, Jingyi Song, and Hao Wang. 2025b. Airrag: Autonomous strategic planning and reasoning steer retrieval augmented generation. *Preprint*, arXiv:2501.10053.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *Preprint*, arXiv:2508.07976.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. 2024. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11143–11156, Bangkok, Thailand. Association for Computational Linguistics.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

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.

Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.

Chen Hu, Haikuo Du, Heng Wang, Lin Lin, Mingrui Chen, Peng Liu, Ruihang Miao, Tianchi Yue, Wang You, Wei Ji, and 1 others. 2025. Step-deepresearch technical report. *arXiv preprint arXiv:2512.20491*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 737–740.

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.

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.

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.

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

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025a. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *Preprint*, arXiv:2509.13305.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025b. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

5131–5157, Vienna, Austria. Association for Computational Linguistics.

Weijie Li, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2025c. Topology-of-question-decomposition: Enhancing large language models with information retrieval for knowledge-intensive tasks. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2814–2833, Abu Dhabi, UAE. Association for Computational Linguistics.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025d. Webthinker: Empowering large reasoning models with deep research capability. *Preprint*, arXiv:2504.21776.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke S. Zettlemoyer, and Scott Yih. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *ArXiv*, abs/2310.01352.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, Chengjun Xie, Botian Shi, Yong Liu, and Yu Qiao. 2025. O^2-searcher: A searching-based agent model for open-domain open-ended question answering. *Preprint*, arXiv:2505.16582.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6355–6384, Vienna, Austria. Association for Computational Linguistics.

Maja Pavlovic. 2025. Understanding model calibration – a gentle introduction and visual exploration of calibration and the expected calibration error (ece). *Preprint*, arXiv:2501.19047.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Gabriele Prato, Jerry Huang, Prasanna Parthasarathi, Shagun Sodhani, and Sarath Chandar. 2024. Do large language models know how much they know? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6054–6070, Miami, Florida, USA. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *Preprint*, arXiv:2509.13309.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Ning Shang, Yifei Liu, Yi Zhu, Li Lyna Zhang, Weijiang Xu, Xinyu Guan, Buze Zhang, Bingcheng Dong, Xudong Zhou, Bowen Zhang, Ying Xin, Ziming Miao, Scarlett Li, Fan Yang, and Mao Yang. 2025. rstar2-agent: Agentic reasoning technical report. *Preprint*, arXiv:2508.20722.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *Preprint*, arXiv:2505.11277.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *Preprint*, arXiv:2503.05592.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025a. Zerosearch: Incentivize the search capability of llms without searching. *Preprint*, arXiv:2505.04588.

Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025b. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *Preprint*, arXiv:2505.16834.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*.

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.

Hongru Wang, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, and Kam-Fai Wong. 2025a. Toward a theory of agents as tool-use decision-makers. *Preprint*, arXiv:2506.00886.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. 2025b. Acting less is reasoning more! teaching model to act efficiently. *Preprint*, arXiv:2504.14870.

Jinpeng Wang, Chao Li, Ting Ye, Mengyuan Zhang, Wei Liu, and Jian Luan. 2026. Icpo: Intrinsic confidence-driven group relative preference optimization for efficient reinforcement learning. *Preprint*, arXiv:2511.21005.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. Webdancer: Towards autonomous information seeking agency. *Preprint*, arXiv:2505.22648.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, Xun Deng, Zhikai Lei, Miao Zheng, Guoteng Wang, Shuo Zhang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, and 2 others. 2025. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *Preprint*, arXiv:2510.18927.

Hongshen Xu, Zichen Zhu, Situo Zhang, Da Ma, Shuai Fan, Lu Chen, and Kai Yu. 2024. Rejection improves reliability: Training llms to refuse unknown questions using rl from knowledge feedback. *Preprint*, arXiv:2403.18349.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2025. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *Preprint*, arXiv:2509.02479.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Preprint*, arXiv:2407.02485.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025a. Parallelsearch: Train your llms to decompose query and search sub-queries in parallel with reinforcement learning. *Preprint*, arXiv:2508.09303.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025b. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1024, Vienna, Austria. Association for Computational Linguistics.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. Sglang: Efficient execution of structured language model programs. *Preprint*, arXiv:2312.07104.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, and 14 others. 2025. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *Preprint*, arXiv:2406.15877.

A Knowledge Boundary and Search Regulation

Search Regulation for Deep-Search Agents. Industrial systems such as Tongyi Deep Research (Li et al., 2025a; Qiao et al., 2025; Li et al., 2025d; Wu et al., 2025) further push end-to-end deep-search capabilities in post-training. Despite these advances, many agents boost accuracy by relying on external evidence as a shortcut, and tool calls keep rising while accuracy gains saturate, suggesting that *how*-to-search improves faster than *when*-to-search calibration. To better balance accuracy and tool cost, recent methods introduce model-side need signals or retrieval-specific shaping rewards to regulate search behavior (Shi et al., 2025; Wang et al., 2025b), but such self-generated supervision can be unreliable and hard to transfer across models and tasks, limiting evaluation of when-to-search behavior across settings.

Knowledge Boundary for Retrieval Decisions. The knowledge boundary separates queries answerable from parametric knowledge from those requiring external information (Li et al., 2025b; Xu et al., 2024). Existing probes are mainly of two types: (i) template-based prompting tests (Prato et al., 2024; Petroni et al., 2019; Li et al., 2025c), and (ii) internal-signal predictors based on hidden states, SAEs, or confidence (Chen et al., 2025b; Zhao et al., 2025b). Because these methods rely on internal cues, they can be miscalibrated and hard to compare across models. Aggregate outcome metrics such as accuracy and tool calls also conflate information need with model capability and task difficulty, making necessary and redundant tool use hard to distinguish. Thus, prior work still lacks a verifiable objective that directly aligns tool invocation with model-specific information need. RSS instead uses protocol-preserving no-search outcomes to estimate a model-specific deficit signal, and evaluates whether realized tool use matches this estimated need.

B Rational Search Score Details and Additional Analyses

In this appendix, we provide additional evidence and sensitivity analyses for the rational search score (RSS), which operationalizes need-use alignment for external information seeking. Concretely, RSS relies on two episode-level signals: an outcome-based deficit indicator S_i estimated by high-diversity no-search sampling, and a task-conditioned adequacy indicator Y_i computed by a hop-conditioned tool-use band. For each instance x_i, we draw K no-search outputs $\{\hat{y}_i^{(k)}\}_{k=1}^{K} \sim \pi(\cdot \mid x_i)$ using high temperature and compute the no-search correctness rate

$$p_i = \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}\left[\mathrm{Corr}\left(\hat{y}_i^{(k)}, g_i\right) = 1\right], \qquad (1)$$
$$S_i = \mathbf{1}[p_i < \tau].$$

Given the realized number of external calls $c_i = |Z_{\mathrm{out}}|$, we define the adequacy indicator by a task-structure proxy (e.g., hop) as

$$Y_i = \mathbf{1}[t_{\min}(h_i) \le c_i \le t_{\max}(h_i)], \qquad (2)$$

and combine them into

$$\mathrm{RSS}_i \triangleq S_i \cdot Y_i + (1 - S_i) \cdot \mathbf{1}[c_i \le t_0],$$
$$\mathrm{RSS} \triangleq \frac{1}{N} \sum_{i=1}^{N} \mathrm{RSS}_i. \tag{3}$$

Here, t_0 is a small slack for no-deficit cases. The remainder of this section validates the reliability of the high-diversity deficit estimation and the adequacy band construction, and reports sensitivity to K and τ.

B.1 High-Diversity No-Search Sampling for Deficit Estimation

RSS requires a model-specific deficit signal to separate information need from realized tool use. Since $W_{\text{int}}(m)$ is a black box, state-level deficit is neither comparable nor practical; we therefore estimate it with an outcome-based, episode-level proxy under a protocol-preserving no-search setting. Concretely, we keep the interaction template unchanged and replace each retrieval return with the same fixed no-evidence stub, `This experiment will NOT return retrieval results; rely solely on your internal knowledge and reasoning.`, so that prompts and execution format are preserved while the external evidence channel carries **no task-specific information**. This comparison is protocol-preserving, but not perfectly matched in a causal sense: once retrieved evidence is removed, the model's subsequent reasoning trajectory and action choices may also change, so a fully identical intervention is not achievable in the current end-to-end agent setting. We therefore adopt the most controlled intervention available under the same agent protocol, rather than presenting it as a clean causal decomposition.

Table 5 reports the bucketed distribution of no-search correctness rate p_i under different decoding temperatures. For 7B, we report representative low/high-temperature settings rather than a full sweep, since the purpose is to illustrate the same degeneracy-to-diversity trend. We highlight two observations. (1) **Low temperature leads to degenerate deficit estimation.** With low temperature, most mass collapses into the extreme buckets, producing many all-wrong or all-correct cases and leaving little resolution near the deficit boundary. (2) **Higher-diversity decoding improves the visibility of borderline cases.** As temperature increases, probability mass shifts from the extremes toward the intermediate buckets, making near-threshold instances more observable when forming S_i. This is exactly the regime needed for deficit estimation: the no-search signal should not be dominated by near-deterministic outputs, but should expose whether an instance is stably answerable from internal knowledge alone. Overall, high-diversity no-search sampling is not merely a probing detail; it is what makes the outcome-based deficit signal practically separable near the threshold. Although the illustration here uses Qwen-family agents, the role of temperature is methodological rather than model-specific: higher-diversity decoding makes near-threshold cases easier to observe whenever repeated no-search sampling is available.

	Unknown		Known	
T	$[0, 0.25)$	$[0.25, 0.5)$	$[0.5, 0.75)$	$[0.75, 1]$
3B				
0.7	838	0	0	162
0.9	752	51	56	141
1.1	757	64	51	128
1.3	727	94	58	121
7B				
0.7	659	29	54	258
1.7	719	27	30	224

Table 5: Bucketed distribution of no-search correctness rate p_i under different decoding temperatures on NQ-1000 (K=4), using the default half-split threshold τ=0.5. With K=4, this threshold labels 0–1 correct samples as deficit and 2–4 as non-deficit. Increasing temperature shifts mass from the extreme buckets toward the intermediate buckets, making near-threshold cases less degenerate and easier to separate when forming S_i. Blocks report Search-R1-3B and Search-R1-7B as the no-search policies.

B.2 Why Confidence Is Not a Reliable Deficit Proxy

Confidence reflects how decisively a model commits to a completion under a given decoding setting, but this is not equivalent to whether task-relevant external evidence is actually missing. Empirically, Fig. 7 provides a broad view across datasets and model scales, showing a persistent confident-but-wrong (CBW) region under protocol-preserving no-search probing: a non-trivial portion of incorrect predictions still receive high confidence. In both Fig. 7 and Fig. 2, the outer contour denotes all incorrect predictions, while the colored inner region marks the subset that is both incorrect and high-confidence. Fig. 2 shows the same phenomenon in two representative settings, with NQ serving as a

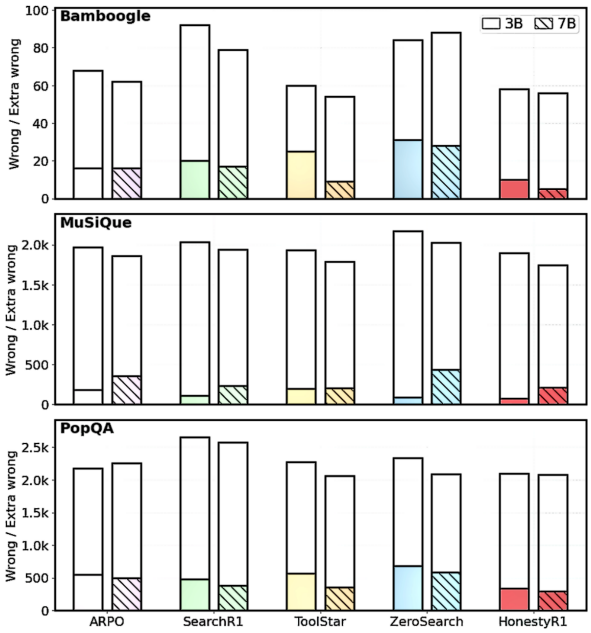

Figure 7: Confident-but-wrong (CBW) errors under protocol-preserving no-search probing across Bamboogle, MuSiQue, and PopQA at both 3B and 7B scales. The outer contour denotes all incorrect predictions, while the colored inner region marks the subset that is both incorrect and high-confidence. A noticeable portion of errors remains in this inner region across datasets and model sizes, showing that confidence alone is not a reliable proxy for whether external information is actually needed.

mostly single-hop reference and 2Wiki providing more explicit compositional structure. Taken together, these results show that high confidence can arise from fluent but unsupported internal completions rather than from genuinely sufficient internal knowledge.

More fundamentally, confidence is difficult to use as a transferable deficit criterion across models and settings. Due to calibration differences and decoding effects, the same confidence value can correspond to different conditional error rates across architectures, checkpoints, or inference hyperparameters, making any single threshold brittle. Confidence also shifts with task difficulty relative to model capability, so the confidence–correctness relationship is not stable enough to serve as a difficulty-aware deficit proxy in general. Accordingly, we estimate deficit from protocol-preserving no-search rollouts via the outcome statistic p_i (and its thresholded label S_i), which directly measures empirical correctness under the same interaction protocol and thus provides a more comparable signal across model scales and difficulty regimes. We do not claim confidence is always uninformative; rather, we find it insufficient as a standalone, comparable deficit proxy across settings.

B.3 Robustness to K and Deficit Threshold Choices

Our deficit estimate is based on protocol-preserving no-search multi-sampling: we compute the no-search correctness rate p_i and threshold it into a deficit label S_i, so a natural question is why we fix $K{=}4$ and use $\tau{=}0.5$. These two choices play different roles in RSS: K determines the sampling variance and the discretization granularity of p_i, whereas τ only specifies a decision boundary on the discrete p_i scale to separate deficit vs. non-deficit, without altering the underlying semantics of internal solvability without external evidence.

Choice of K: statistical stability and diminishing returns. For a fixed instance x_i under a fixed no-search decoding setting, let $X_{i,k} \in \{0, 1\}$ indicate whether the k-th no-search sample is correct, and denote the per-sample correctness probability by $p_T \triangleq \Pr(X_{i,k} = 1)$. We estimate p_T by $\hat{p}_i \triangleq \frac{1}{K}\sum_{k=1}^{K} X_{i,k}$, which satisfies $\mathbb{E}[\hat{p}_i] = p_T$. Thus, increasing K does not change the estimation target, but only reduces sampling noise. Modeling $\{X_{i,k}\}_{k=1}^{K}$ as repeated Bernoulli trials under a standard conditional-independence approximation (conditioning on the prompt, policy, and decoding hyperparameters), we obtain

$$\mathrm{Var}(\hat{p}_i) = \frac{p_T(1-p_T)}{K} \;\le\; \frac{1}{4K}, \qquad (4)$$

where the upper bound is attained at $p_T = 0.5$, i.e., the maximally ambiguous regime; near-deterministic cases ($p_T \approx 0$ or 1) yield almost all-wrong or all-correct samples and hence much smaller variance. Equivalently, $\sigma_{\max}(\hat{p}_i) \le \frac{1}{2\sqrt{K}}$, giving

$$\begin{aligned} K = 2 &: 0.353, \quad K = 4 \quad : 0.250, \\ K = 8 &: 0.177, \quad K = 16 \quad : 0.125. \end{aligned}$$

so doubling K yields only a $1/\sqrt{K}$ stability gain while sampling cost grows linearly. This trade-off is salient in our setting, where no-search probing is run at dataset scale and across multiple temperatures and checkpoints. Moreover, label flips concentrate near the threshold. Let $S_i = \mathbf{1}[\hat{p}_i < \tau]$ and $m \triangleq |p_T - \tau|$. By Hoeffding's inequality (Hoeffding, 1963),

$$\Pr\big[\,|\,\hat{p}_i - p_T\,| \ge m\,\big] \;\le\; 2\exp(-2Km^2) \qquad (5)$$

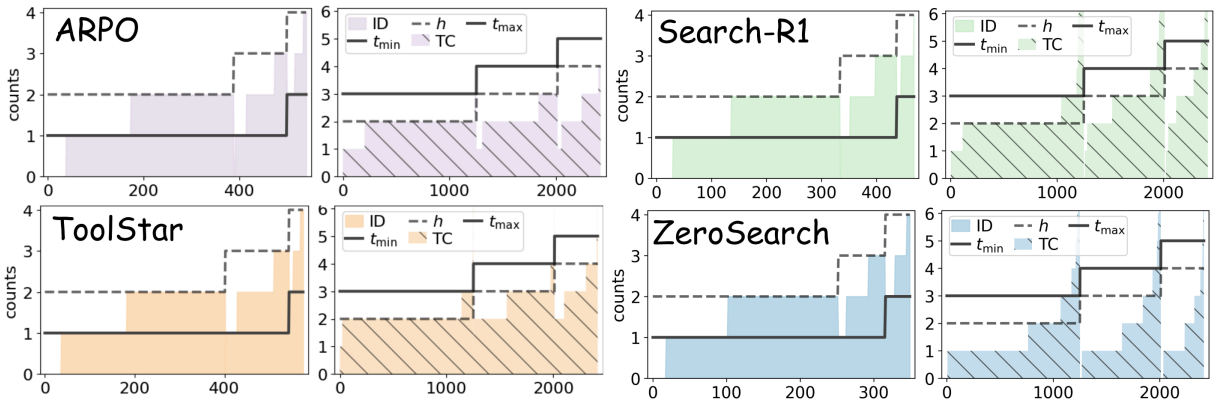

Figure 8: Sanity check for the hop-conditioned tool-use band on MuSiQue (3B). MuSiQue provides gold sub-questions, so for each instance we use the number of gold sub-questions as the structure cue h. The left panel plots an empirical information-deficit count (ID), obtained by checking under no-search probing whether each gold sub-question remains answerable from the model's internal knowledge, together with the corresponding h. The right panel compares the realized tool-call count TC against the band $[t_{\min}(h),\, t_{\max}(h)]$, where $t_{\min}(h) = \max(1, \lfloor h/2 \rfloor)$ and $t_{\max}(h) = h + 1$. Across methods, higher-structure instances tend to induce larger deficit counts, while successful trajectories are broadly consistent with the resulting adequacy band rather than a single fixed call budget.

Therefore, instances with a non-trivial margin have exponentially small flip probability even for small K.

Finally, $K{=}4$ induces a clean, protocol-compatible discretization: since $\sum_{k=1}^{K} X_{i,k} \in \{0, 1, 2, 3, 4\}$, $\hat{p}_i = \frac{1}{4}\sum_{k=1}^{4} X_{i,k}$ takes values on a 0.25 grid, which aligns with the bucket intervals in Table 5. We therefore fix $K{=}4$ as a compute–stability trade-off that is sufficient to expose borderline uncertainty without paying the linear overhead of larger-K sampling.

Choice of deficit threshold τ. The threshold τ only discretizes the outcome-based statistic $\hat{p}_i$ into $S_i = \mathbf{1}[\hat{p}_i < \tau]$; it does not change the quantity being estimated by $\hat{p}_i$. We set $\tau = 0.5$ as a natural, protocol-compatible half-split boundary: with $K{=}4$, this labels an instance as deficit when fewer than half of the no-search samples are correct (0–1 hits), and as non-deficit otherwise (2–4 hits), treating the tie case $(2/4)$ as non-deficit for a conservative split. Sensitivity to τ is inherently concentrated on borderline instances. Let p_T denote the (unknown) per-sample no-search correctness probability and let $m \triangleq |p_T - \tau|$ be the margin. Eq. (5) implies that the probability of flipping S_i decays exponentially in Km^2; thus, changing τ primarily affects near-threshold cases, while instances with p_T far from τ remain stable.

B.4 Adequacy Band as a Verifiable Filter Rather Than an Oracle Budget

The hop-conditioned band is a verifiable tolerance filter for tool use, rather than an oracle that predicts an optimal call budget. Given the realized call frequency $c_i = |Z_{\text{out}}|$, the band checks whether c_i falls within a structure-conditioned range. This is more robust than a point target: a single $\texttt{out}(\cdot)$ call may return variable amounts of evidence, so call counts primarily measure interaction cost rather than the number of atomic facts obtained.

We validate this construction on MuSiQue, where gold sub-questions provide an explicit structure cue with minimal confounding. For each instance, we use the number of gold sub-questions as h, derive the band by $t_{\min}(h) = \max(1, \lfloor h/2 \rfloor)$ and $t_{\max}(h) = h + 1$, and compute an empirical information-deficit count (ID) by checking under no-search probing whether each gold sub-question remains answerable from internal knowledge. As shown in Fig. 8, higher-structure instances tend to induce larger deficit counts, while successful trajectories are broadly consistent with the resulting band; under-evidence and over-triggering are more likely to fall below $t_{\min}(h)$ or above $t_{\max}(h)$. Therefore, we use the band as a lightweight, verifiable gate for separating under- and over-triggering, rather than as an estimate of the minimal evidence needed for each instance.

B.5 Empirical Robustness of RSS

This subsection examines the empirical robustness of the adequacy-band component in RSS. Unlike the earlier K/τ analysis, which concerns deficit estimation, the analyses below focus on whether the main need-use alignment conclusions remain stable when the practical band construction is varied through the structure proxy h and the tolerance ratio r.

Practical adequacy-band construction beyond hop annotations. In hop-unavailable settings, the adequacy band must be instantiated from a practical structure proxy rather than exact hop-level ground truth. Our claim is therefore not that a single band choice is uniquely correct, but that the band can be anchored to dataset-side structure or difficulty cues instead of being treated as a free hyperparameter. In our benchmarks, this issue is limited in practice because most datasets already provide usable cues beyond explicit hop labels. For example, deep-search benchmarks often include explicit difficulty signals such as GAIA Level 1/2/3 (Mialon et al., 2023), WebWalkerQA Easy/Medium/Hard (Zheng et al., 2025), Humanity's Last Exam NS/CE/SF (Phan et al., 2025), and official step recommendations in xbench-DeepSearch (Chen et al., 2025a). In more standard QA settings, general QA (e.g., NQ (Kwiatkowski et al., 2019), SimpleQA (Wei et al., 2024), and TriviaQA (Joshi et al., 2017)) is typically single-hop, while multi-hop benchmarks usually already provide hop annotations, decomposition cues, or explicit structure labels. We therefore instantiate the adequacy band from dataset-side cues whenever available, and treat h as a practical coarse-structure proxy rather than an oracle quantity. As a result, the applicability of RSS does not depend narrowly on the availability of exact hop annotations: in practice, it remains usable as long as reasonable dataset-side structure or difficulty cues are available.

Robustness to adequacy-band scale h. Table 6 reports RSS under different adequacy-band scales h with fixed $r = 0.50$. In our main setting, h is not taken as the exact hop count of each instance, but as a practical dataset-side structure proxy: PopQA is treated as mostly single-hop (h=1), Bamboogle as harder multi-hop (h=2), and 2Wiki/MuSiQue as structure-aware multi-hop settings (roughly $h \in [2, 4]$). Table 6 is therefore a deliberate robustness test: we intentionally sweep h away from these practical choices to test whether the main conclusions are tied to a narrow band specification. We highlight two observations. (1) **RSS remains stable in a reasonable range around the practical band choice.** For moderate values ($h = 1, 2, 4$), the relative ordering of methods is largely preserved across datasets and model scales, showing that RSS does not rely on exact per-instance hop annotations. (2) **Clear degradation appears only under severely mismatched band choices.** When h becomes unrealistically large ($h = 8, 16$), RSS drops sharply across datasets because the band is severely mismatched to the actual task structure and no longer separates adequate from mismatched tool use. Thus, even without strict hop-level ground truth, RSS still captures a meaningful need-use alignment trend under practical structure proxies.

Dataset	Method	$h = 1$	$h = 2$	$h = 4$	$h = 8$	$h = 16$
3B						
Bam	ARPO	0.648	0.648	0.560	0.352	0.352
Bam	Search-R1	0.712	0.712	0.648	0.312	0.304
Bam	Rational-R1	0.744	0.744	0.736	0.280	0.256
Bam	ToolStar	0.576	0.576	0.560	0.432	0.432
MuSiQue	ARPO	0.909	0.909	0.818	0.101	0.091
MuSiQue	Search-R1	0.922	0.922	0.866	0.200	0.111
MuSiQue	Rational-R1	0.949	0.949	0.945	0.273	0.051
MuSiQue	ToolStar	0.912	0.912	0.906	0.149	0.089
PopQA	ARPO	0.829	0.829	0.308	0.171	0.171
PopQA	Search-R1	0.846	0.846	0.553	0.184	0.172
PopQA	Rational-R1	0.844	0.844	0.758	0.167	0.156
PopQA	ToolStar	0.832	0.832	0.474	0.171	0.168
7B						
Bam	ARPO	0.512	0.512	0.440	0.488	0.488
Bam	Search-R1	0.608	0.608	0.552	0.424	0.392
Bam	Rational-R1	0.760	0.760	0.704	0.248	0.248
Bam	ToolStar	0.496	0.496	0.528	0.632	0.576
MuSiQue	ARPO	0.857	0.857	0.833	0.163	0.120
MuSiQue	Search-R1	0.889	0.889	0.815	0.397	0.123
MuSiQue	Rational-R1	0.974	0.974	0.667	0.025	0.020
MuSiQue	ToolStar	0.892	0.892	0.824	0.457	0.210
PopQA	ARPO	0.811	0.811	0.410	0.184	0.184
PopQA	Search-R1	0.805	0.805	0.576	0.296	0.198
PopQA	Rational-R1	0.920	0.920	0.337	0.058	0.058
PopQA	ToolStar	0.825	0.825	0.595	0.334	0.237

Table 6: RSS sensitivity under different adequacy-band scales h with fixed $r = 0.50$. RSS remains stable under practical band choices, while very large h values make the band mismatched to the actual task structure and lead to clear degradation.

Robustness to tolerance ratio r. Table 7 reports RSS under different tolerance ratios r. Here r controls the strictness of the adequacy band around the structure-conditioned reference: smaller r yields a tighter band, while larger r yields a looser one. We use $r = 0.50$ as the default choice because it provides a natural middle ground between an overly strict adequacy test and an overly permissive one. We highlight two observations. (1) **RSS is stable under practical choices of r.** Across $r \in \{0.25, 0.50, 0.75\}$, the main comparison is generally preserved under practical choices, al-

Dataset	Method	$r = 0.25$	$r = 0.50$	$r = 0.75$
3B				
PopQA	ARPO	0.829	0.836	0.829
PopQA	Search-R1	0.846	0.862	0.846
PopQA	Rational-R1	0.844	0.884	0.844
PopQA	ToolStar	0.832	0.843	0.832
Bam	ARPO	0.648	0.696	0.560
Bam	Search-R1	0.712	0.744	0.648
Bam	Rational-R1	0.744	0.776	0.736
Bam	ToolStar	0.576	0.696	0.560
MuSiQue	ARPO	0.909	0.931	0.571
MuSiQue	Search-R1	0.922	0.943	0.753
MuSiQue	Rational-R1	0.949	0.967	0.863
MuSiQue	ToolStar	0.912	0.953	0.769
2Wiki	ARPO	0.645	0.684	0.647
2Wiki	Search-R1	0.682	0.653	0.644
2Wiki	Rational-R1	0.668	0.687	0.667
2Wiki	ToolStar	0.645	0.649	0.717
7B				
PopQA	ARPO	0.811	0.815	0.811
PopQA	Search-R1	0.805	0.809	0.805
PopQA	Rational-R1	0.920	0.846	0.920
PopQA	ToolStar	0.825	0.810	0.825
Bam	ARPO	0.512	0.568	0.440
Bam	Search-R1	0.608	0.672	0.552
Bam	Rational-R1	0.760	0.760	0.704
Bam	ToolStar	0.496	0.568	0.528
MuSiQue	ARPO	0.857	0.887	0.723
MuSiQue	Search-R1	0.889	0.923	0.741
MuSiQue	Rational-R1	0.974	0.924	0.397
MuSiQue	ToolStar	0.892	0.903	0.784
2Wiki	ARPO	0.622	0.663	0.631
2Wiki	Search-R1	0.617	0.583	0.627
2Wiki	Rational-R1	0.610	0.683	0.651
2Wiki	ToolStar	0.573	0.557	0.620

Table 7: RSS robustness under different tolerance ratios r. Here r controls the strictness of the adequacy band around the structure-conditioned reference, with $r = 0.50$ used as the default middle setting. Across practical choices of r, the relative comparison is generally preserved, indicating that RSS is not driven by a brittle band-width choice; overly loose settings may degrade high-structure cases.

though overly loose settings can degrade RSS on some high-structure datasets. (2) **Changing r mainly affects band strictness rather than the qualitative conclusion.** Tighter or looser bands shift the adequacy criterion, and extreme looseness can hurt high-structure cases, but the default middle setting avoids this degeneracy while preserving the main need-use alignment trend. Overall, these results support the use of $r = 0.50$ as a practical default rather than a brittle single-threshold choice.

C Training Details

Here we provide implementation-level details for reproducing RSS-guided training, including the tool protocol, training data construction and difficulty cues, validation design for training-dynamics tracking, and optimization settings. We also summarize RSS-specific engineering choices that make the reward verifiable and practical under GRPO.

C.1 MDP View and Reward Observability

Induced interaction process. We view tool-augmented reasoning as an episodic controlled process induced by tool execution. Consider a single-model agent system $\Sigma(m_\theta) = (E(m_\theta), \{\pi_\theta\})$ with typed actions $A = \{\texttt{int}(\cdot), \texttt{out}(\cdot), \texttt{stop}(\cdot)\}$ and information states $s_t = (o_t, I_t)$, where I_t is the accumulated information set and o_t is the observation derived from the latest information fragment. At step t, the policy samples an action $a_t \sim \pi_\theta(\cdot \mid s_t)$. If $a_t \neq \texttt{stop}(\cdot)$, the environment executes the corresponding tool and returns an information fragment $z_t \sim \mathrm{Exec}(\cdot \mid s_t, a_t; W)$, which is merged via $I_{t+1} = \mathrm{Upd}(I_t, z_t)$ and yields the next state $s_{t+1} = (\psi(z_t), I_{t+1})$. The episode terminates at the first T such that $a_T = \texttt{stop}(y)$, inducing an episodic process $\mathcal{M}(m_\theta) = (\mathcal{S}, A, P, R, \gamma)$ with trajectory distribution

$$p_\theta(\tau) = \rho_E(x) \prod_{t=0}^{T-1} \pi_\theta(a_t \mid s_t)\, P(s_{t+1} \mid s_t, a_t),$$
$$\tau = (x, s_0, a_0, z_0, \ldots, s_{T-1}, a_{T-1}, z_{T-1}, a_T). \tag{6}$$

In our setting, $P(s_{t+1} \mid s_t, a_t)$ is induced by tool execution: the environment first samples $z_t \sim \mathrm{Exec}(\cdot \mid s_t, a_t; W)$ and then deterministically updates $s_{t+1} = (\psi(z_t), \mathrm{Upd}(I_t, z_t))$. Therefore, retrieved spans $\{z_t\}$ are part of the transition outcome (environment-side randomness), while the policy controls the generated reasoning, search queries, whether to invoke $\texttt{out}(\cdot)$, and when to stop.

Verifiable terminal reward from observable rollout summaries. Our reward is terminal and computed from an episode-level summary that is directly readable from the finalized rollout. Let $Z_{\mathrm{out}} \triangleq \biguplus_{t<T}\{z_t : a_t = \texttt{out}(\cdot)\}$ be the multi-set of external fragments and $c_i \triangleq |Z_{\mathrm{out}}|$ be the realized external call count (we use c_i to avoid overloading r_t). To estimate whether external access is needed for an instance x_i, we run a protocol-preserving no-search probe that turns off external evidence while keeping the same interaction template: we draw K no-search samples, compute an empirical correctness rate $\hat{p}_i$, and threshold it to a deficit label S_i. This probe is executed once per instance under a fixed decoding setting and cached, so S_i can be reused across GRPO rollouts/groups without repeated no-search sampling.

RSS is then computed from (S_i, c_i) and a task proxy, e.g., hop, for band checking, enabling need-use alignment to be verified at the episode level.

A step-level deficit notion at intermediate states s_t is neither observable nor comparable because $W_{\text{int}}(m_\theta)$ is a black box, and state-wise deficit evaluation would require expensive counterfactual rollouts. During GRPO, we treat retrieved spans as environment outcomes and mask `<information>` contents from the loss (including policy-gradient and KL terms), updating only LLM-generated tokens. This prevents trivial copying of tool outputs and focuses gradients on decision- and reasoning-based behaviors that govern *when* to trigger $\texttt{out}(\cdot)$.

Concretely, the terminal reward combines answer correctness, a lightweight format check, and RSS:

$$r(x_i, \tau_i) = r_{\text{acc}}(y_i, g_i) + \lambda\,\text{RSS}_i + \lambda_f\, r_{\text{fmt}}(\tau_i),$$

where RSS_i depends only on (S_i, c_i) (and h_i for band checking) and is verifiable from the finalized rollout.

Division of labor: SFT learns actions, RL learns decisions. We adopt an SFT→RL pipeline with a fixed tool interface and execution protocol. SFT primarily teaches *how* to act by enforcing executable formatting and protocol compliance (parseable calls and proper termination), ensuring rollouts are well-formed and tool execution is reliable. RL then learns *when* to act under the same protocol by updating π_θ to decide whether to invoke $\texttt{out}(\cdot)$ or stop, using a verifiable terminal reward that includes RSS. This matches our motivation: outcome-only rewards conflate necessity-aligned triggering with shortcut search, whereas RSS makes need-use alignment observable at the episode level and thus learnable by RL.

C.2 GRPO Training Objective for RSS-Guided Search

Building on the episode-level reward in Sec. C.1, we train a search-augmented policy $\pi_\theta(\cdot \mid x; R)$ with GRPO (Shao et al., 2024). For each prompt x, we sample a group of G interleaved rollouts $\{y_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot \mid x; R)$. Since each rollout contains both model-generated tokens and retrieved evidence, we mask out retrieved spans and update only LLM-generated tokens. Let $M_{i,t} \in \{0, 1\}$ denote this LLM-token mask. The objective is

$$\max_{\theta} \quad \mathbb{E}_{x,\{y_i\}_{i=1}^{G}} \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{t} M_{i,t}\, \ell_{i,t}(\theta) \right] - \beta D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big), \tag{7}$$

where

$$\ell_{i,t}(\theta) = \min \Big(\rho_{i,t}(\theta)\hat{A}_i, \; \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i \Big). \tag{8}$$

where

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t}; R)}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t}; R)}. \tag{9}$$

Here $\hat{A}_i$ is the group-relative advantage computed from the RSS-guided episode rewards and broadcast to valid LLM-generated tokens. The KL term is computed on the same valid-token mask $M_{i,t}$, so retrieved evidence spans do not contribute to either the PPO term or the KL penalty.

C.3 Training Data, Difficulty Cues, and Evaluation Suite

Building on Sec C.1, where RSS is a terminal and verifiable signal computed from observable rollout summaries, we describe the training ingredients that make RSS-guided optimization reproducible and diagnosable. Starting from Qwen2.5-3B/7B-Instruct (Qwen et al., 2025), we obtain Rational-R1 (3B/7B) with an SFT→RL pipeline. SFT teaches *how* to use tools via a parseable protocol and typed actions $A = \{\texttt{int}(\cdot), \texttt{out}(\cdot), \texttt{stop}(\cdot)\}$, while RSS-guided GRPO teaches *when* to search (invoke $\texttt{out}(\cdot)$ vs. stop). For stability and need-aware learning, we train RL with a curriculum (Stage1→Stage2) and compare it with a mixed-all schedule, using the same held-out validation pool to isolate the effect of objectives and schedules. We follow the copyright and licensing terms associated with all datasets and open-source code.

SFT cold-start: sources, filtering, and protocol normalization. Our cold-start SFT set is assembled from multiple tool-interaction sources, including curated protocol templates and demonstrations from O^2-Searcher (Mei et al., 2025), SimpleDeepSearcher (Sun et al., 2025b), and R1-Search (Song et al., 2025), as well as reproduced tool-use traces from prior search-RL agents (ARPO/ToolStar) (Dong et al., 2025c,b). Notably, the SFT portions used by ARPO and ToolStar contain highly similar search-demonstration traces under a near-identical tool protocol; we therefore treat them as a single compatible source and merge them into a single pool. We apply dataset-level hygiene to make the cold-start set stable and non-redundant:

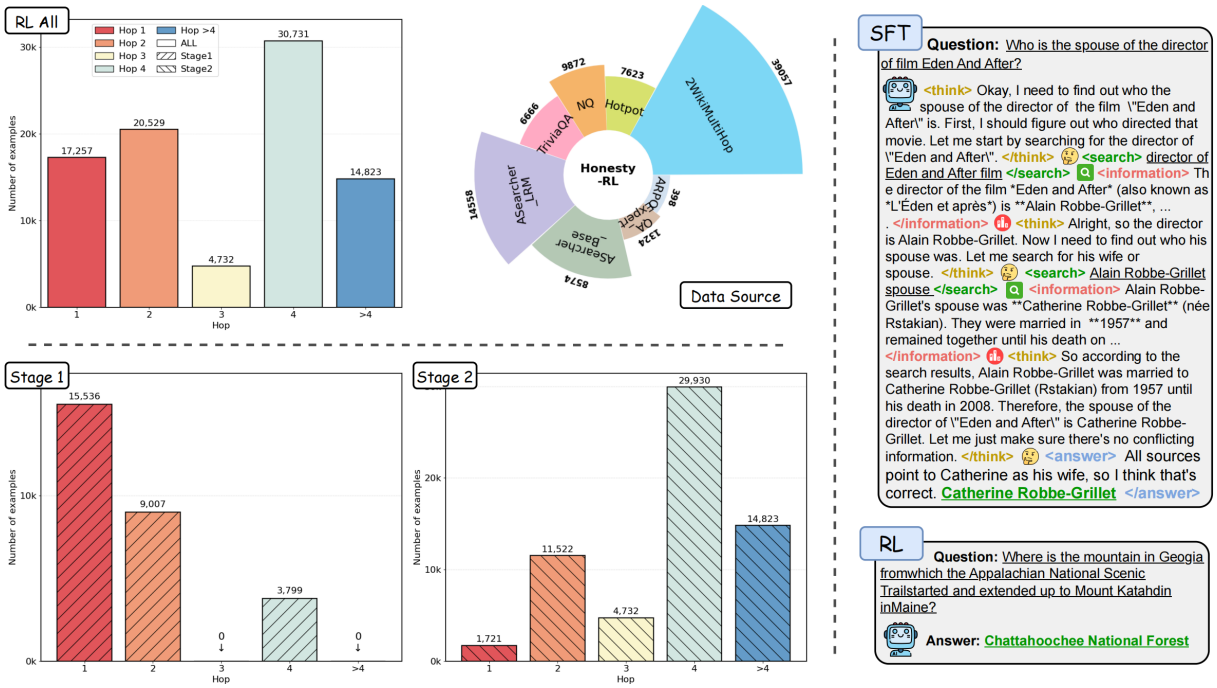

Figure 9: Training data overview. **Left:** hop distribution of the full RL training pool (RL-ALL), with 17,257 / 20,529 / 4,732 / 30,731 / 14,823 examples in the 1/2/3/4/>4 hop bins, respectively. The Stage1$\rightarrow$Stage2 curriculum shifts the distribution from lower-hop examples in Stage-1 (15,536 / 9,007 / 0 / 3,799 / 0) to higher-hop and longer-horizon examples in Stage-2 (1,721 / 11,522 / 4,732 / 29,930 / 14,823). The middle panel summarizes the source mixture of the RL pool. **Right:** representative examples of the unified executable tool protocol used in both SFT cold-start and RL rollouts.

we filter noisy items, remove near-duplicate trajectories, and use QwQ-32B as a difficulty filter to exclude overly trivial instances (little signal beyond format imitation) and excessively hard ones (often unstable traces). We then normalize all examples into a unified executable protocol with `<think>`, `<search>`, `<information>`, and `<answer>` fields and typed actions $A = \{\texttt{int}(\cdot), \texttt{out}(\cdot), \texttt{stop}(\cdot)\}$. After cleaning and normalization, we obtain 9,981 SFT examples. About 9k of them are drawn from the same tool-SFT lineage used by ARPO and ToolStar, while the remainder only covers closely related search-interaction formats. While smaller than the $\geq$20k cold-start data used in some pipelines, this stage is deliberately scoped as a stable tool-use warm start rather than a larger supervised advantage: it only needs to ensure protocol compliance (parseable calls and proper termination), so that RSS-guided RL can focus on learning *when* to trigger $\texttt{out}(\cdot)$ rather than being dominated by format or parse failures.

RL data. Fig. 9 (left) summarizes the sources and hop distribution of our RL training set. The RL corpus mixes multi-hop QA (2WikiMultiHopQA, HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022)), general QA (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017)), agent-style reasoning traces (ASearcher-LRM/Base (Gao et al., 2025), QA-Expert[2], and ARPO-style data), together with high-difficulty deep-search tasks. For datasets that provide evidence lists, we run a retrieval reachability check under the training-time retriever protocol (2018 Wikipedia, E5, top-3 passages) and discard samples whose annotated evidence is unreachable or inconsistent, avoiding annotation–retriever mismatch that would otherwise contaminate RSS-related signals. Datasets without evidence lists rely solely on answer-level supervision.

We attach an explicit difficulty cue to each instance. When hop metadata is available, we use dataset-provided labels; for structurally explicit cases (e.g., `bridge/comparison` questions in 2Wiki), we standardize them as 2-hop. For high-difficulty samples without hop labels, we infer hop using Qwen3-thinking and manually filter obvious failures. RL uses a two-stage curriculum: Stage-1 contains 25,344 examples, dominated by lower-hop instances, with a small fraction of higher-hop data

[2]https://huggingface.co/khaimaitien/qa-expert-7B-V1.0

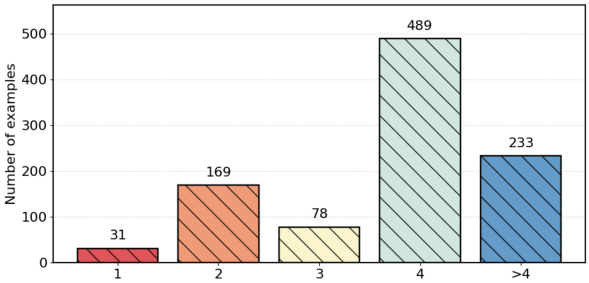

Figure 10: Validation pool for training-dynamics tracking. The pool contains 1,000 held-out examples, distributed across hop bins as 31 / 169 / 78 / 489 / 233 for 1/2/3/4/>4 hops, respectively. This balanced design prevents the tracked curves from being dominated by trivial low-hop cases and supports controlled comparison across reward objectives and training schedules.

retained. In contrast, Stage-2 contains 62,728 examples, emphasizing higher-hop and deep-search instances, with limited low-hop retention. We compare the Stage1→Stage2 curriculum against a mixed-all schedule using the same held-out validation pool. Stage-1 and Stage-2 are curriculum subsets sampled from the same deduplicated RL pool and may overlap; their counts therefore do not sum to the pool size. Overall, the RL pool comprises 68,072 instances (Fig. 9).

Validation set for training dynamics. To keep evaluation affordable during RL, we track learning curves on a fixed 1,000-example validation pool sampled from our evaluation benchmarks. The pool is stratified across QA sources and balanced in hop difficulty (Fig. 10), preventing the curves from being dominated by trivial low-hop instances. We reuse the same pool for all reward objectives (`acc-only`, `acc+conf`, `acc+RSS`) and schedules (Cur vs. All), enabling controlled comparisons. This is particularly important for RSS: since adequacy is conditioned on hop via the band check $t_{\min}(h) \leq c_i \leq t_{\max}(h)$, a hop-balanced pool makes the measured accuracy–TC trade-off reflect *when*-to-search behavior rather than sampling bias. The pool is used only for tracking training dynamics; all final results are evaluated on the full benchmark splits.

C.4 Implementation Details

We implement RSS-guided training with SkyRL (Cao et al., 2025) using GRPO. Following prior tool-RL practice, we mask tool-call `<information>` contents from the loss to avoid overfitting to tool outputs; only LLM-generated tokens (reasoning and action tokens) are optimized. All main training runs are conducted on 4× NVIDIA A100 GPUs (80GB), while the overhead study additionally reports a pipelined setting with one extra GPU for RSS generation. We use a context length of 4096 and set the maximum generation length to 4096. For RL, we sample 4 rollouts per prompt and cap each episode to at most 8 external interactions. The global batch size is 128 for 3B and 64 for 7B (micro-batch size 4 per GPU). We use AdamW with learning rate 1×10^{-6} and train for one epoch. Unless otherwise specified, the `acc-only` objective uses a weighted sum of accuracy and a format-validity term (0.9/0.1). For objectives with an auxiliary signal (`acc+conf` and `acc+RSS`), we use a unified weighting (0.7/0.2/0.1), where the 0.2 term is instantiated as confidence/RSS_i.

RSS/Confidence-guided training. Both RSS and confidence objectives require an auxiliary no-search signal computed under a fixed decoding protocol. We fix the deficit threshold to $\tau = 0.5$ and estimate no-search correctness by sampling with temperature $T = 1.1$ and $K = 4$. For each question x_i, we draw K no-search answers with a fixed random seed, compute the empirical correctness rate $\hat{p}_i$, and discretize it into the deficit label $S_i = \mathbf{1}[\hat{p}_i < \tau]$. Crucially, S_i is defined at the *question level* (not per rollout): it depends only on the no-search probing of x_i, so it is shared by all GRPO rollouts generated from the same x_i and can be reused across groups. During training, we cache the auxiliary no-search statistics and pair them with the finalized rollout summary. For RSS, this combines the cached deficit label S_i with the realized call count c_i and band indicator Y_i; for the confidence baseline, the cached confidence statistic is used as its auxiliary reward term. This keeps the auxiliary signal protocol-consistent and verifiable from the finalized rollout. Detailed wall-clock overhead is reported separately in Sec. C.6.

Prompt template. All SFT and RL rollouts follow a unified system prompt that enforces the executable tool protocol and the structured fields used by our evaluator.

System Prompt

You are an honest model. Answer the given question. You must conduct reasoning inside `<think>...</think>` every time you receive new information. After reasoning, if you truly lack the necessary knowledge, you may call a search engine using `<search>query</search>`; results will appear between `<information>...</information>`. Only use the search engine when you genuinely lack the knowledge. If no further external knowledge is needed, provide the final answer directly inside `<answer>...</answer>`, without detailed explanations. For example: `<answer>Beijing</answer>`. Question:

Resource Setting	Method	Wall-clock Time
Same 4×A100 Budget		
4 train GPUs	3B acc-only	34 h
3 train GPUs + 1 RSS-gen GPU	3B RSS	38 h
4 train GPUs	7B acc-only	72 h
3 train GPUs + 1 RSS-gen GPU	7B RSS	79 h
With One Extra GPU		
4 train GPUs	3B acc-only	baseline
4 train GPUs + 1 RSS-gen GPU	3B RSS	baseline + 14 min
4 train GPUs	7B acc-only	baseline
4 train GPUs + 1 RSS-gen GPU	7B RSS	baseline + 23 min

Table 8: Wall-clock overhead of RSS-guided training under two optimized resource settings. RSS adds a protocol-preserving no-search@K probe to compute the cached deficit label S_i, implemented as an empty-generation batch rather than a second full rollout/update pass. Under the same 4×A100 budget, wall-clock time increases from 34h to 38h for 3B and from 72h to 79h for 7B. With one additional GPU for RSS generation, the added cost drops to 14 minutes for 3B and 23 minutes for 7B.

C.5 Fairness & Implementation Alignment

Our SFT→RL diagnosis (e.g., Fig. 6 and quadrant breakdowns) requires that baseline SFT checkpoints be obtained and evaluated under the same execution semantics; otherwise, observed differences could be attributed to protocol, retriever, or TC/RSS accounting discrepancies rather than RL. Therefore, for each RL baseline (e.g., ARPO/ToolStar), we *only reproduce its SFT checkpoint* using the released codebase and original public SFT data, while keeping our tool protocol and evaluator unchanged. We run SFT reproductions three times with different random seeds and report the mean results for baseline-SFT in our behavioral analyses.

We emphasize that the compared methods do *not* follow an identical SFT+multi-stage RL pipeline. Our method uses a lightweight tool-interaction SFT warm start (9,981 examples) intended to provide a stable tool-use cold start rather than substantial additional task supervision. About 9k examples are drawn from the same tool-SFT lineage used by ARPO and ToolStar, while the remainder only covers closely related search-interaction formats; thus, our warm start is smaller than the full ARPO/ToolStar SFT stage and is not a larger supervised advantage. For baselines such as Search-R1 and ZeroSearch, we follow their published training recipes rather than forcing all methods into a single unified pipeline. Accordingly, we present our comparison as *protocol-faithful* to each baseline's published recipe rather than as a fully unified retraining setup.

For all methods (reproduced baseline-SFT, released baseline-RL checkpoints, and our models), we use the same unified execution setup—including the same retriever configuration, tool protocol parsing, and tool-cost/RSS computation—so that TC/RSS/quadrant statistics are implementation-aligned and directly comparable. Baseline RL checkpoints are taken from official releases; we do **not** re-train their RL stages.

C.6 Wall-clock overhead of RSS-guided training

RSS introduces additional no-search@K computation in principle, since each training instance requires a protocol-preserving no-search probe to obtain the cached deficit label S_i. However, in our implementation this extra cost is lightweight rather than equivalent to a second full training loop. Concretely, the no-search signal is computed with an *empty-generation batch* under the same prompting protocol, instead of re-running a full with-tool rollout and update pass. Moreover, this auxiliary computation is overlapped with the main SkyRL training pipeline: while the current batch performs rollout generation, reward postprocessing, and optimization, the no-search probing for the next batch is executed on a dedicated RSS-generation GPU and cached in advance. As a result, RSS adds one lightweight auxiliary generation stream rather than a second full RL update path.

Table 8 reports wall-clock time under two optimized resource settings. Under the same 4×A100 budget, the `acc-only` baseline uses all four GPUs for training, whereas RSS uses three training GPUs plus one RSS-generation GPU, increasing wall-

clock time from 34h to 38h for 3B (+11.7%) and from 72h to 79h for 7B (+9.7%). With one additional GPU, RSS keeps the same four training GPUs and moves the no-search probing entirely to the extra GPU, reducing the added wall-clock cost to **only 14 minutes for 3B and 23 minutes for 7B.** Therefore, although RSS requires extra no-search computation, the practical overhead remains modest because the added work is lightweight and largely hidden by overlap/pipelining.

D Experimental Setup

Unless otherwise specified, experiments are conducted with two model variants, Qwen-2.5-3B-Instruct and Qwen-2.5-7B-Instruct (Qwen et al., 2025).

D.1 Datasets

Datasets. We evaluate Rational-R1 on seven knowledge-intensive QA benchmarks and two deep-search benchmarks, grouped by task type: (1) **General QA**—NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); (2) **Multi-Hop QA**—HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023); (3) **Deep-Search**—GAIA (Mialon et al., 2023) and XBench (Chen et al., 2025a). The main experiments focus on the seven knowledge-intensive QA benchmarks, while GAIA and XBench are used only for supplementary deep-search evaluation. We focus most behavioral analyses on 2Wiki and NQ, while controlled sanity checks for the hop-conditioned adequacy band use MuSiQue, whose hop labels and gold sub-questions provide explicit structure cues. We briefly summarize each benchmark below:

- **Natural Questions (NQ)** (Kwiatkowski et al., 2019): open-domain QA from real user queries, evaluated on Wikipedia-grounded answers.
- **TriviaQA** (Joshi et al., 2017): large-scale open-domain trivia QA with evidence documents, emphasizing retrieval-heavy factual questions.
- **PopQA** (Mallen et al., 2023): entity-centric QA constructed from Wikidata triples with popularity metadata, probing long-tail factual knowledge.
- **HotpotQA** (Yang et al., 2018): Wikipedia-based multi-hop QA requiring evidence aggregation across multiple documents.
- **2WikiMultiHopQA** (Ho et al., 2020): multi-hop QA combining Wikidata and Wikipedia, covering diverse compositional reasoning patterns. For 2WikiMultiHopQA, we use the 2,500-example dev subset, whose hop distribution matches that of the full dev set.
- **MuSiQue** (Trivedi et al., 2022): compositional multi-hop QA with explicit decomposition cues (hop/sub-questions), used for controlled band sanity checks.
- **Bamboogle** (Press et al., 2023): hand-crafted challenging multi-hop QA (up to four hops) designed to stress compositional reasoning.
- **GAIA** (Mialon et al., 2023): deep-search benchmark for general AI assistants with real-world questions requiring web browsing and multi-step tool use.
- **XBench** (Chen et al., 2025a): profession-aligned agent benchmark evaluating end-to-end tool-using capability on real-world tasks.

Difficulty / structure annotations (h_i). RSS uses a per-instance structure cue h_i only to instantiate the hop-conditioned adequacy band $[t_{\min}(h_i), t_{\max}(h_i)]$. We emphasize that h_i is a *coarse proxy for task structure* (for verifiable and comparable band checking), not an oracle for the minimal evidence required. We set h_i as follows: (i) for **General QA** benchmarks (NQ/TriviaQA/PopQA), we set h_i=1; (ii) for **HotpotQA** and **Bamboogle**, we use h_i=2 following their dataset descriptions; (iii) for **2WikiMultiHopQA**, we infer h_i from its compositional construction (how atomic sub-questions are combined); (iv) for **MuSiQue**, we use the provided hop label / number of sub-questions; (v) for **Deep-Search** benchmarks (GAIA/XBench), which do not provide reliable hop-style annotations, we use a fixed proxy h_i=8 when a structure cue is required.

D.2 Baselines

We compare Rational-R1 against three groups of baselines spanning prompting-only inference, retrieval-augmented inference, and RL-based search agents. (1) **Inference without Retrieval**: Direct inference and Chain-of-Thought (CoT) (Wei

et al., 2023); (2) **Inference with Retrieval**: Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) and IRCoT (Trivedi et al., 2023); (3) **RL-Based Search Agents**: Search-R1 (Jin et al., 2025a), ZeroSearch (Sun et al., 2025a), ToolStar (Dong et al., 2025b), and ARPO (Dong et al., 2025c).

These methods represent distinct design choices for search-augmented reasoning. Search-R1 trains multi-turn retrieval behavior through RL with explicit search triggers; ZeroSearch replaces live search with a simulated retriever and trains search behavior under a curriculum; ToolStar bootstraps multi-tool interaction from synthesized trajectories and often improves performance with increased tool usage; ARPO improves *how* agents use tools by branching rollouts at high-entropy post-tool steps and assigning branch-level credit. For each baseline, we follow its published recipe as closely as possible, prioritizing faithful reproduction over forcing all methods into a single training pipeline. For the SFT→RL behavioral diagnosis in our analysis, we additionally reproduce baseline SFT checkpoints under our unified evaluator; implementation details are provided in Appendix C.5.

We also include OTC (Wang et al., 2025b) as a **contextual low-call reference**. Because OTC is not publicly released, we cannot perform a protocol-aligned reimplementation or run the same calibration-style analyses under our evaluator. We therefore report OTC only using the results provided in the original paper, and interpret it as a contextual low-call operating point rather than a protocol-aligned benchmark for necessity-aware triggering.

D.3 Retrieval Setup and Tool-Cost Accounting

Retriever setup. For all Wikipedia-based benchmarks, we use the same local retriever across all retrieval-based methods to ensure comparable evidence access and tool costs. The retriever is built on the Wikipedia 2018 corpus released in FlashRAG (Jin et al., 2025b) and uses E5-base-v2 (Wang et al., 2024) as the dense retriever. We deploy the retriever as an external service with sglang (Zheng et al., 2024) and return top-k=3 passages per query.

Deep-search tasks. For GAIA and XBench, we evaluate Rational-R1 in an open-web setting rather than under the local Wikipedia retriever. During inference, we use Serper[3] as the interface for Google Search and retrieve up to 10 webpages per query, following prior deep-search setups (Li et al., 2025d). Because these benchmarks differ from the Wikipedia-based QA setting in evidence source, search interface, and task format, we report them as supplementary deep-search evaluations rather than as directly pooled results with the seven QA benchmarks. For deep search, we also report results from stronger models including QwQ (Team, 2024), DeepSeek-R1 (Guo et al., 2025), GPT-4o (Hurst et al., 2024), and o1-preview (Hurst et al., 2024), together with Qwen3-32B (Yang et al., 2025), as contextual comparisons only. Since these systems differ in model scale, tool protocol, and compute budget, we do not treat them as strict head-to-head baselines.

Tool calls (TC). We measure tool cost by the number of external retrieval invocations (call frequency):

$$\mathrm{TC} = c_i = |Z_{\mathrm{out}}|.$$

Concretely, c_i counts how many times the agent calls the retrieval service for an instance x_i. Each retrieval query counts as one call, regardless of how many passages or webpages are returned.

Cover-EM. In the training-dynamics plots, we report cover-EM as a relaxed exact-match metric for long-form agent outputs. Unlike strict EM, which requires the normalized predicted answer to exactly match the gold answer, cover-EM counts an output as correct if the normalized final response contains a gold answer string or otherwise covers the reference answer under the dataset evaluator. Formally, for prediction $\hat{y}_i$ and gold answer set $\mathcal{G}_i$, we compute

$$\mathrm{CoverEM}_i = \mathbf{1}\Big[\exists g \in \mathcal{G}_i : \ \mathrm{norm}(g) \text{ is a substring of } \mathrm{norm}(\hat{y}_i)\Big].$$

with the same answer normalization used by the evaluation script. We use cover-EM in training curves because search agents often produce longer final responses, where the correct short answer may appear inside a full sentence rather than as an exact standalone string.

D.4 Metrics and Inference Settings

Accuracy metrics. We report answer accuracy using **exact match (EM)** throughout the paper unless

[3] https://serper.dev/

otherwise specified. For training-dynamics tracking (Fig. 5), we additionally report **Cover-EM**, defined in Sec. D.3, because RL rollouts may include extra formatting tokens or short justifications around the answer. In contrast, the main tables use EM for strict comparison across methods and benchmarks. We use Cover-EM for curves because RL rollouts may include extra formatting tokens or short justifications around the answer. In contrast, the main tables use EM for strict comparison across methods and benchmarks.

RSS accounting. RSS is computed from two episode-level signals: a deficit label S_i and an adequacy label Y_i. We obtain S_i from a *protocol-preserving no-search probe*: for each x_i we sample K no-search answers, compute an empirical correctness rate $\hat{p}_i$, and threshold it at $\tau = 0.5$:

$$S_i = \mathbf{1}[\hat{p}_i < \tau].$$

We compute Y_i from the realized call count c_i via a hop-conditioned adequacy band. For instances with structure proxy h_i, we use $t_{\min}(h) = \max(1, \lfloor h/2 \rfloor)$ and $t_{\max}(h) = h + 1$. For no-deficit cases ($S_i = 0$), we allow a slack $t_0 = 1$ to tolerate a single "safety" call. Unless otherwise stated, we follow the main paper's RSS definition and quadrant mapping when reporting RSS/quadrant statistics.

Decoding settings. For all with-tool evaluations, we use the same decoding configuration across methods under our unified tool protocol. Following Search-R1, we use a temperature $T = 0.9$ and nucleus sampling with top-$p = 0.95$. For no-search probing (deficit estimation), we use $K = 4$ and threshold $\tau = 0.5$, with a higher temperature to encourage diversity: $T = 1.1$ for 3B and $T = 1.3$ for 7B (top-$p = 0.95$). We fix the random seed to **42** for with-tool evaluation to ensure reproducibility. For no-search probing, we draw K stochastic samples using distinct random seeds (e.g., a fixed base seed with per-sample offsets), so that $\{\hat{y}_i^{(k)}\}_{k=1}^{K}$ is diverse yet reproducible; consequently, S_i is deterministic given the probing protocol and can be cached and reused across rollouts and objectives.

E Supplementary Analyses

Sec. 5 focuses on 3B to isolate the core behavioral mechanism: accuracy-driven search-RL can improve *how* search is used, but does not reliably calibrate *when* to trigger `out`$(\cdot)$, whereas RSS-guided GRPO makes need-use alignment verifiable and suppresses redundant triggering. To examine whether the same picture persists at a larger scale and under complementary evaluation views, this section provides four supplementary analyses. We first extend the main behavioral diagnosis to 7B through training dynamics and a quadrant-level breakdown on 2Wiki under the same evaluator and tool-cost accounting. We then report deep-search performance on GAIA and XBench to assess generalization beyond the Wikipedia-retriever setting. Finally, we include the auxiliary efficiency proxy Δ as a compact operating-point summary of average accuracy relative to tool-use cost. Across these supplementary views, the qualitative conclusions remain consistent with the main analysis: scaling alone does not guarantee better search calibration, while RSS-guided training yields more stable and better-aligned search behavior.

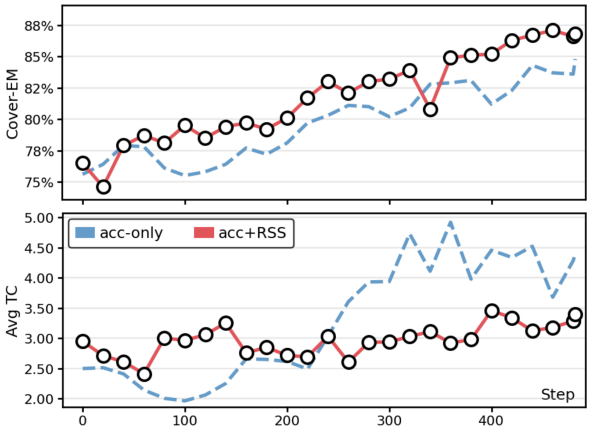

Figure 11: Training dynamics of 7B models under Stage2 curriculum. We compare `acc-only` and `acc+RSS`, reporting Cover-EM (top) and average tool calls (TC; bottom) over steps.

E.1 7B Training Dynamics under RSS-Guided GRPO

To complement the main analysis (Sec. 5), we report a 7B-scale trend verification under the same Stage2 curriculum and evaluation protocol as the 3B dynamics. We fine-tune Rational-R1 (7B) with GRPO and compare two objectives: `acc-only` and `acc+RSS`. Fig. 11 plots Cover-EM (top) and average tool calls (TC; bottom) over training steps. We highlight two consistent observations. (1) **`acc+RSS` preserves a late-stage accuracy advantage.** Both objectives improve rapidly at the beginning, but `acc-only` shows earlier saturation, whereas `acc+RSS` continues to improve and reaches a higher final Cover-EM. This mirrors the 3B

trend in Fig. 5, suggesting that rewarding need-use alignment encourages adequate evidence acquisition on deficit cases (S=1), creating better conditions for using retrieved evidence effectively. (2) **acc+RSS stabilizes tool-use behavior at 7B.** In the TC curves, acc-only exhibits a noticeable upward drift in later training, while acc+RSS remains in a tighter range with smoother evolution. This supports the same diagnosis as in the main content: scaling the backbone can improve *how* to use search, but, by itself, does not calibrate *when* to trigger it. By making the adequacy of tool use verifiable via the hop-conditioned band, RSS-guided GRPO introduces an explicit constraint on redundant triggering, yielding a more favorable balance between accuracy and TC at convergence.

E.2 7B Quadrant-Level Diagnosis of Search Triggering

The main analysis suggests an objective gap: optimizing end accuracy alone does not reliably align tool use with information need. To trace this gap to a concrete behavioral source at 7B, Table 9 reports the RSS quadrant breakdown on the 2WikiMultiHop dev subset, comparing ToolStar (SFT vs. RL) and Rational-R1 (SFT, RL-ACC, RL-RSS). The same failure pattern remains clear at the larger scale: for baseline agents, most excess tool calls arise from unnecessary triggering in $(-,+)$ rather than missed searches in $(+,-)$, so TC keeps increasing without producing more need-aligned behavior. For ToolStar, SFT→RL further amplifies this mismatch: the dominant mass remains in $(-,+)$, while $(-,-)$ stays nearly empty, indicating that RL does not systematically suppress redundant triggering. Our controlled ablation makes the missing signal more explicit at 7B. RL-ACC can change *how* search is used, but still leaves unnecessary triggering largely intact, whereas RL-RSS makes *when* to search verifiable and shifts behavior from $(-,+)$ toward need-aligned abstention $(-,-)$, substantially reducing redundant tool calls. Together with Fig. 6, this breakdown shows that the dominant behavioral failure at 7B is still cost-heavy over-triggering, and that RSS provides the missing signal needed to suppress it.

E.3 Deep Search Performance

To extend evaluation beyond Wikipedia QA and assess performance in a realistic open-web setting, we further test Rational-R1 on two deep search benchmarks, GAIA and XBench, using a web search engine during inference. Table 10 also reports results from stronger ≥32B reasoning models as contextual references; since these systems differ in model scale, tool protocols, and compute budgets, we do not make head-to-head comparisons. The referenced large-scale results are reported under their original evaluation protocols, whereas Rational-R1 is evaluated as a tool-augmented agent with web search access. Overall, Rational-R1 exhibits non-trivial deep-search capability despite its small backbone, suggesting that the learned search-triggering behavior can transfer beyond the controlled Wikipedia retriever setting.

Method	(+,+)	(+,-)	(-,-)	(-,+)	TC
ToolStar					
SFT	1566	101	1	832	2.55
RL	1391	153	2	954	3.62
Rational-R1					
SFT	1580	89	4	827	2.56
RL-ACC	1408	166	1	925	3.31
RL-RSS	1422	92	330	621	1.87

Table 9: Quadrant breakdown on 2WikiMultiHop (7B). The first sign denotes information deficit S and the second denotes tool-use adequacy Y: $(+,+)$ need-aligned search, $(+,-)$ under-evidence, $(-,-)$ need-aligned abstention, and $(-,+)$ over-triggering.

Methods	GAIA	XBench-DR	Avg
Direct Reasoning (≥32B)			
Qwen3-32B-thinking	14.9	14.0	14.5
DeepSeek-R1-32B	14.2	10.0	12.1
QwQ-32B	18.9	10.7	14.8
GPT-4o	17.5	18.0	17.8
DeepSeek-R1-671B	25.2	32.7	29.0
Rational-R1			
3B	10.3	7.0	8.7
7B	19.4	12.0	15.7

Table 10: Deep search on GAIA and XBench. Scores (%) are averaged across the two benchmarks (Avg; higher is better). ≥32B models are reported as contextual reference.

E.4 Auxiliary Efficiency Proxy Δ

Efficiency proxy Δ. We additionally report an efficiency proxy

$$\Delta = \frac{\text{Avg Acc}}{\text{Avg TC}},$$

as one of the most intuitive ratio-form summaries of answer accuracy relative to tool-use cost. We include it as a compact supplementary analysis because it provides a quick operating-point view of

Method	3B			7B		
	Avg Acc	Avg TC	Δ	Avg Acc	Avg TC	Δ
OTC-GRPO◇	0.366	**1.008**	**0.363**	0.361	**0.990**	**0.365**
Search-R1	0.354	1.946	0.182	0.421	2.332	0.181
ZeroSearch	0.368	1.467	0.251	0.382	1.625	0.235
ToolStar	0.465	1.833	0.254	0.510	2.875	0.177
ARPO	0.446	1.565	0.285	0.477	1.809	0.264
Rational-R1	**0.485**	1.549	0.313	**0.520**	1.653	0.315

Table 11: Compact comparison of average accuracy, average tool calls, and the auxiliary efficiency proxy $\Delta = \text{Avg Acc}/\text{Avg TC}$. We report Δ only as a compact summary rather than a standalone criterion, **since ratio-only comparisons can favor low-call operating points without establishing better necessity-aware triggering.** OTC-GRPO◇ is therefore included as a contextual low-call reference only, reported from the original paper.

how much answer accuracy is obtained per unit of tool use, which is often helpful for comparing methods with different search budgets. However, we treat Δ only as an auxiliary proxy rather than a standalone evaluation criterion. A ratio of this form can be mechanically inflated by extremely low tool volume even when answer quality remains poor (e.g., $0.20/0.10 = 2.0$). It can also penalize methods that spend more calls on deficit-heavy instances where additional evidence is genuinely needed. Therefore, our main conclusions rely on the joint view of answer accuracy, tool-call behavior, and RSS-based need-use alignment, rather than on Δ alone.

A Compact Ratio-Form View via Δ. Table 11 reports average accuracy, average tool calls, and the auxiliary efficiency proxy Δ for representative search-RL baselines together with the contextual low-call reference OTC-GRPO. From these results, we highlight two observations. (1) **Lower-call operating points can attain a larger Δ without establishing better necessity-aware triggering.** This is reflected by OTC-GRPO, which achieves the highest Δ at both scales, yet remains substantially lower than Rational-R1 in absolute average accuracy. (2) **Rational-R1 maintains the strongest overall position among the main RL baselines.** At both 3B and 7B, Rational-R1 attains the highest average accuracy while keeping tool use comparable to, or lower than, several strong RL baselines, yielding a competitive Δ without relying on indiscriminate call suppression.

F More Case Studies of Rational-R1

To better understand the *when*-to-search behaviors learned by Rational-R1, we present additional qualitative case studies, using Search-R1 as the primary baseline. All trajectories follow our unified tool protocol: model reasoning, queries, retrieved evidence, and final outputs are marked by <think>, <search>, <information>, and <answer>, respectively, which facilitates consistent visualization of *rational* and *irrational* search behaviors. Following the RSS quadrant definition, the first sign denotes whether an information deficit is estimated (S), and the second sign denotes tool-use behavior relative to that need: $(+,+)$ need-aligned search, $(+,-)$ under-evidence, $(-,-)$ need-aligned abstention, and $(-,+)$ over-triggering. We use these cases to illustrate how Rational-R1 differs from Search-R1 in *when* it searches, and how this behavior evolves from 3B to 7B due to stronger internal knowledge and more effective evidence use. Representative examples are provided in Tables 12, 13, 14, and 15.

Case 1 (Need-aligned search vs. under-evidence). As shown in Table 12, the question requires identifying the directors of two films and comparing their birth dates, creating a clear information deficit (S=1). Rational-R1 falls into the $(+,+)$ quadrant: it retrieves the necessary intermediate facts (director $\rightarrow$ birth date), stays within the hop-conditioned adequacy band (Y=1), and produces the correct answer. By contrast, Search-R1 falls into the $(+,-)$ quadrant: it stops after an early noisy retrieval, forms an incorrect director chain, and fails due to insufficient evidence. This case shows that rational search is not captured by tool-call count alone; the key is whether search is triggered and sustained when the task actually requires additional information.

Case 2 (Correctness can mask over-triggering). Tables 13 and 14 compare two trajectories that both end with the correct answer. Rational-R1 falls into the $(-,-)$ quadrant: after obtaining sufficient evidence, it avoids unnecessary additional search and answers with only two targeted queries, consistent with need-aligned abstention under $c_i \leq t_0$. Search-R1, however, falls into the $(-,+)$ quadrant: it also reaches the correct final answer, but uses six tool calls and includes irrelevant or invalid search steps. This case shows why terminal accuracy alone is not diagnostic of *when*-to-search behavior: a correct answer may still be produced through an inefficient, over-triggered trajectory. RSS addresses this limitation by separating need-use alignment from mere terminal correctness.

Case 3 (Scaling improves internal sufficiency).

As shown in Table 15, Rational-R1-3B falls into the $(+,+)$ quadrant: it is deficit-labeled (S=1), performs a two-step need-aligned search (performer $\rightarrow$ nationality), and stays within the adequacy band (Y=1). Rational-R1-7B, in contrast, shifts to the $(-,-)$ quadrant: it is judged non-deficit (S=0) and completes the task with only a single lightweight confirmation call under $c_i \leq t_0$. This suggests that scaling can reduce *necessary* tool use when internal knowledge becomes sufficient, while RSS enables a capability-aware comparison of *when*-to-search behavior across model sizes.

Question: Which film has the director who was born later, Tongues Of Flame or Glory Alley? ($h = 4$)

Ground Truth: Tongues Of Flame

Rational-R1 → Q1: Need-aligned search (S=1, Y=1)

<think> I need to find out which film has the director who was born later between "Tongues of Flame" and "Glory Alley". I'll start by searching for the directors of these films. </think> <search> Who directed the film "Tongues of Flame"? </search>

<information> Doc 1(Title: "Jean-Jacques Annaud") Jean-Jacques Annaud Jean-Jacques Annaud (born 1 October 1943) is a French film director, · · · </information>

<think> I found out that "Tongues of Flame" is a 1936 film directed by Wesley Ruggles. Now I'll search for the director of "Glory Alley". </think> <search> Who directed the film "Glory Alley"? </search>

<information> Doc 1(Title: "Glory Alley") Glory Alley Glory Alley is a 1952 musical drama film directed by Raoul Walsh. · · · </information>

<think> Now I know that "Glory Alley" is a 1952 film directed by Raoul Walsh. I'll compare the birth years of Wesley Ruggles and Raoul Walsh to determine which film was made later. </think> <search> The birth date of Wesley Ruggles (film director) </search>

<information> Doc 1(Title: "Wesley Ruggles") in Glendale, California, near his brother, Charles Ruggles. For his contributions to the motion picture industry, he has a star on the Hollywood Walk of Fame at 6400 Hollywood Boulevard. · · · </information>

<think> From the information found, Wesley Ruggles, the director of "Tongues of Flame," was born on June 11, 1889. Now I'll find out when Raoul Walsh, the director of "Glory Alley," was born. </think> <search> birth date of Raoul Walsh (film director) </search>

<information> Doc 1(Title: "Raoul Walsh") York as Albert Edward Walsh to Elizabeth T. Bruff, the daughter of Irish Catholic immigrants, and Thomas W. Walsh, · · · </information>

<think> From the information found, Raoul Walsh, the director of "Glory Alley," was born on March 11, 1887. Comparing the birth years of Wesley Ruggles (June 11, 1889) and Raoul Walsh (March 11, 1887), it's clear that Raoul Walsh was born earlier. Therefore, the film made later, with the director born later, is "Tongues of Flame". </think>

<answer> Tongues of Flame </answer> $TC = 4$ ✔

Search-R1 → Q2: Under-evidence (S=1, Y=0)

<think> To determine which film has the director who was born later, Tongues Of Flame or Glory Alley, I will follow these steps: 1. Identify the directors of the two films. 2. Determine which director was born later. Let's start with step 1: identifying the directors of the two films. </think> <search> Tongues Of Flame director </search>

<information> Doc 1(Title: "Pirjo Honkasalo") Pirjo Honkasalo Pirjo Irene Honkasalo (born 22 February 1947) is a Finnish film director · · · </information>

<think> From the search results, we can see that the director of "Tongues Of Flame" is Pekka Lehto, and the director of "Glory Alley" is Tonga Ta'ele. Now, let's move to step 2: determining which director was born later. Pekka Lehto was born on 10 January 1950. Tonga Ta'ele was born on 30 August 1951. Since Tonga Ta'ele was born later than Pekka Lehto, the director of "Glory Alley" was born later. Now, I will provide the final answer. </think>

<answer> Glory Alley </answer> $TC = 1$ ✘

Table 12: Case study 1 (2WikiMultiHopQA, h=4): under-evidence vs. need-aligned search. Both agents face an estimated information deficit (S=1). Rational-R1 issues *deficit-aware* retrieval and stays within the hop-conditioned adequacy band (Y=1), collecting the necessary intermediate facts and answering correctly. In contrast, Search-R1 makes fewer calls but falls outside the adequacy band (Y=0): it stops after an early, noisy hit and commits to a wrong chain. This case links back to our central claim: tool-call frequency alone is not a reliable indicator of rational search—RSS explicitly verifies whether search is *needed* and whether the acquired evidence is *adequate*, separating need-aligned querying from cost-cutting under-evidence.

Question: Which film has the director born later, Back To The Door or Charles, Dead Or Alive? ($h = 4$)

Ground Truth: Charles, Dead Or Alive

Rational-R1 → Q3: Need-aligned abstention (S=0, limited search)

<think> To solve this question, I need to find out the birthdates of the directors of "Back To The Door" and "Charles, Dead Or Alive" and then compare them to determine which film has the director born later. Step 1: Identify the director of "Back To The Door". Step 2: Identify the director of "Charles, Dead Or Alive". Step 3: Compare the birthdates of the directors and determine which film has the director born later. Let's start with Step 1: Step 1: Identify the director of "Back To The Door". I'll use a query to search for information about the director of "Back To The Door" </think> <search> Director of "Back To The Door" </search> <information> Doc 1(Title: "Door to Door (film)") Door to Door (film) Door to Door is a 2002 television film · · · directed by Steven Schachter, · · · </information>

<think> Based on the search results, "Back To The Door" is a 1917 American silent comedy-drama film directed by David Smith. Step 2: Identify the director of "Charles, Dead Or Alive". I'll use a query to search for the birthdate of the director of "Charles, Dead Or Alive": </think> <search> Director of "Charles, Dead Or Alive" and their birthdate </search> <information> Doc 1(Title: "Charles, Dead or Alive") Charles, Dead or Alive Charles, Dead or Alive (French: ""Charles mort ou vif"") is a 1970 Swiss drama film directed by Alain Tanner. Produced in reaction to the Protests of 1968, · · · </information>

<think> Based on the search results, "Charles, Dead Or Alive" is a 1970 film directed by Alain Tanner. Step 3: Compare the birthdates of the directors and determine which film has the director born later. David Smith was born in 1874, while Alain Tanner was born in 1947. Since Alain Tanner is a much more recent director, the film "Charles, Dead Or Alive" has the director born later. Therefore, the film that has the director born later is "Charles, Dead Or Alive". </think> <answer> Charles, Dead Or Alive </answer> $TC = 2$ ✔

Table 13: Case study 2-A (2WikiMultiHopQA, h=4): **need-aligned abstention after sufficient evidence.** Rational-R1 reaches the correct answer with only two targeted searches and then avoids further tool use, reflecting limited search after sufficient evidence rather than repeated triggering.

Question: Which film has the director born later, Back To The Door or Charles, Dead Or Alive? ($h = 4$)

Ground Truth: Charles, Dead Or Alive

Search-R1 → Q4Over-triggering$S$=0, c_i>t_0

<think> To determine which film has the director born later, Back To The Door or Charles, Dead Or Alive, I will follow these steps: 1. Identify the directors of the two films. 2. Find the birth dates of the directors. 3. Compare the birth dates to determine which director was born later. · · · </think>
<search> Back To The Door director </search> <information> Doc 1(Title: "Door to Door (film)") Door to Door (film) Door to Door is a 2002 television film about Bill Porter, · · · </information>

<think> From the search results, we can see that "Back To The Door" is not a real film. Let's move on to the second film, "Charles, Dead Or Alive." </think>
<search> Charles, Dead Or Alive director </search> <information> Doc 1(Title: "Charles, Dead or Alive") Charles, Dead or Alive Charles, Dead or Alive (French: ""Charles mort ou vif"") is a 1970 Swiss drama film directed by Alain Tanner. Produced in reaction to the Protests of 1968, it describes the mid-life crisis of a businessman who decides to drop out of mainstream · · · </information>

<think> From the search results, we can see that the director of "Charles, Dead Or Alive" is Alain Tanner. Now, let's find the birth dates of the directors of both films. </think> <search> Alain Tanner birth date </search> <information>Doc 1(Title: "Alain Tanner") of Paris displeased him; he described it as ""cutthroat."" Between 1960 and 1968, · · · </information>

<think> From the search results, we can see that Alain Tanner was born on 6 December 1929. Now, let's find the birth date of the director of "Back To The Door." </think> <search> Back To The Door director birth date </search> <information> Doc 1(Title: "Door to Door (film)") Door to Door (film) · · · </information>

<think> From the search results, we can see that the director of "Back To The Door" is Steven Schachter, and his birth date is not provided. Now, we compare the birth dates of the two directors: - Alain Tanner: 6 December 1929 - Steven Schachter: Not provided Alain Tanner was born on 6 December 1929, which is earlier than the birth date of Steven Schachter. Therefore, Steven Schachter was born later. My previous action is invalid. If I want to search, I should put the query between <search> and </search>
<information> Doc 1(Title: "And–or tree") And–or tree An · · · </information>

<think> From the search results, we can see that Steven Schachter was not born later than Alain Tanner. Therefore, I need to re-evaluate the information. From the previous steps, we have: - Alain Tanner was born on 6 December 1929. - Steven Schachter's birth date is not provided. · · · Therefore, I will search again to find Steven Schachter's birth date. </think> <search> Steven Schachter birth date </search> <information> Doc 1(Title: "Harry Schachter") Harry Schachter Harry Schachter FRSC (born 25 February 1933 in Vienna, Austria) is · · · </information>

<think> From the search results, we can see that Steven Schachter's birth date is not provided. Therefore, I will directly provide the answer without further searching. </think> <answer> Charles, Dead Or Alive </answer> (lucky hit) $TC = 6$ ✔

Table 14: Case study 2-B (2WikiMultiHopQA, h=4): **correctness can mask over-triggering.** Search-R1 also reaches the correct final answer, but uses six tool calls and includes irrelevant or invalid steps, matching the $(-, +)$ quadrant. This illustrates correct-but-inefficient search: terminal accuracy can hide redundant triggering, while RSS exposes the mismatch between information need and tool use.

Question: What nationality is the performer of song When The Stars Go Blue? ($h = 2$)

Ground Truth: American

Rational-R1-3B → Q1: Need-aligned search (S=1, Y=1)

<think> I need to find out the nationality of the performer of the song "When the Stars Go Blue". I'll start by searching for it. </think> <search> who performed the song "When the Stars Go Blue"? </search>

<information> Doc 1(Title: "When the Stars Go Blue") When the Stars Go Blue ""When The Stars Go Blue"" is a popular alternative country song composed and originally performed by solo artist and former Whiskeytown band member Ryan Adams. · · · </information>

<think> I found out that the song "When the Stars Go Blue" was performed by Ryan Adams. Now I'll search for his nationality. </think> <search> nationality of Ryan Adams (musician)? </search>

<information> Doc 1(Title: "Ryan Adams") Ryan Adams David Ryan Adams (born November 5, 1974) is an American singer-songwriter, musician, multi-instrumentalist, record producer and poet. He is best known for his solo career, · · · </information>

<think> I found out that Ryan Adams is an American musician. Now I can conclude that the nationality of the performer of "When the Stars Go Blue" is American. </think>

<answer> American </answer> $TC = 2$ ✔

Rational-R1-7B → Q3: Need-aligned abstention (S=0, $c_i \leq t_0$)

<think> To solve this question, I need to find out the nationality of the performer of the song "When The Stars Go Blue". I will follow these steps: 1. Identify the performer of the song "When The Stars Go Blue". 2. Determine the nationality of the performer. Let's start with step 1: 1. Identify the performer of the song "When The Stars Go Blue". I'll use a query to search for the performer of "When The Stars Go Blue": </think> <search> Performer of the song "When The Stars Go Blue" </search>

<information> Doc 1(Title: "When the Stars Go Blue") When the Stars Go Blue ""When The Stars Go Blue"" is a popular alternative country song composed and originally performed by solo artist and former Whiskeytown band member Ryan Adams. It was first released with his album ""Gold"" on September 25, 2001. · · · </information>

<think> From the search results, I can see that the performer of the song "When The Stars Go Blue" is Ryan Adams. 2. Determine the nationality of the performer. Ryan Adams is an American singer and songwriter. Therefore, the nationality of the performer of the song "When The Stars Go Blue" is American. </think>

<answer> American </answer> $TC = 1$ ✔

Table 15: Case study 3 (2WikiMultiHopQA-style fact lookup, h=2): **scaling improves internal sufficiency.** Rational-R1-3B falls into the $(+,+)$ quadrant: it is deficit-labeled (S=1), performs need-aligned search within the adequacy band (Y=1), and answers correctly. Rational-R1-7B shifts to the $(-,-)$ quadrant: it is non-deficit (S=0) and stops after a single allowed safety call ($c_i \leq t_0$), reflecting stronger internal knowledge and evidence use. This illustrates that scaling can reduce necessary tool use, while RSS makes such *when*-to-search differences directly comparable.