LEARNING TO SEARCH AND SEARCHING TO LEARN: JOINT KNOWLEDGE INTERNALIZATION VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Integrating large language models (LLMs) with external search engines has significantly enhanced their ability to solve knowledge-intensive tasks. While search-integrated LLMs optimized via reinforcement learning (RL) excel at learning when and how to retrieve information, they typically treat retrieved text strictly as read-only context. To ensure training stability, these retrieved tokens are often masked from the policy-gradient objective, which inherently prevents models from internalizing the factual content. However, directly unmasking these tokens introduces severe noise and destabilizes the early stages of optimization. To resolve this masking dilemma, we propose LfSearch, an end-to-end RL framework that shifts the paradigm from merely learning to search toward searching to learn. LfSearch jointly optimizes search behavior and knowledge internalization through a unified dual-objective loss. By introducing an adaptive masking curriculum, the model is permitted to absorb external facts only after mastering reliable tool use. Furthermore, we apply asymmetric advantage normalization to ensure factual content is strictly reinforced rather than penalized. Across seven single-hop and multi-hop QA benchmarks, LfSearch successfully circumvents the severe performance degradation typically caused by naive unmasking. Empowered by this internalized knowledge, it consistently outperforms baselines like Search-R1, demonstrating that learned facts directly translate to superior dynamic search performance. Finally, evaluations on knowledge-retention sets explicitly confirm that LfSearch effectively distills retrieved facts into parametric memory, enabling the model to genuinely learn from its environment.

1 INTRODUCTION

Large language models have demonstrated remarkable capabilities in natural language understanding and complex reasoning (OpenAI, 2024; Guo et al., 2025; Du et al., 2025). Despite these achievements, their performance on knowledge-intensive tasks remains constrained by hallucinations and static parametric memory (Mallen et al., 2023; Asai et al., 2024). To mitigate these limitations, LLMs are increasingly integrated with external retrieval systems (Lewis et al., 2020; Nakano et al., 2021). Recent work has progressed from static Retrieval-Augmented Generation (RAG) pipelines toward search-augmented LLMs trained with reinforcement learning, where outcome-based rewards teach models to interleave internal reasoning with multi-turn search interactions (Jin et al., 2025; Chen et al., 2025; Wang et al., 2025b).

However, current RL frameworks for search-augmented LLMs largely remain confined to learning to search. To stabilize policy optimization, representative methods mask externally retrieved tokens and compute gradients only over model-generated reasoning and tool calls. This design treats factual evidence as read-only context: the model can learn when and how to retrieve, but the retrieved knowledge itself is never internalized into its parameters. Consequently, the model cannot directly turn the rich evidence encountered during interaction into parametric knowledge. This restricted objective may limit the learning signal after reliable search behavior has emerged and may cause the policy to retrieve information it has already encountered.

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

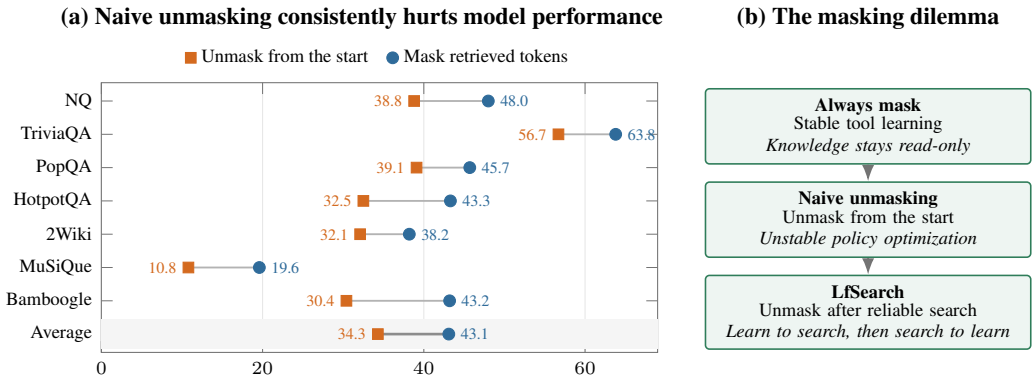

Figure 1: (a) Exact Match (%) of Search-R1 with and without retrieved-token masking on Qwen2.5-7B-base. (b) LfSearch activates retrieved-token learning after reliable search behavior emerges.

Fully exploiting this external knowledge, however, creates a fundamental masking dilemma. Keeping retrieved tokens masked stabilizes RL but prevents knowledge internalization. Naively removing the mask floods the objective with massive external text, creating an overwhelmingly noisy environment that destabilizes the early stages of policy optimization. As illustrated in Figure 1, Search-R1 explicitly studies this alternative and reports that removing retrieved-token masking reduces performance on all benchmarks, with average exact match falling from 43.1% to 34.3% on Qwen2.5-7B-base (Jin et al., 2025). However, this failure does not imply that retrieved knowledge cannot be learned. Instead, it demonstrates that indiscriminate unmasking is an unstable optimization strategy.

To overcome this limitation, we present **L**earning **f**rom **Search** (**LfSearch**), an end-to-end RL framework that moves search-integrated LLMs from merely learning to search toward searching to learn. LfSearch jointly optimizes search behavior and knowledge internalization through a unified dual-objective policy gradient. Rather than permanently masking retrieved evidence or optimizing it indiscriminately, our framework controls when retrieved tokens enter the optimization objective. LfSearch introduces an adaptive masking curriculum to address the instability that motivates conventional loss masking. During early training, optimization is confined to model-generated tokens so that the policy first learns reliable tool-use behavior. Once the sampled trajectories exhibit stable and effective search behavior for a given query, the knowledge internalization objective is activated. We further distinguish the two optimization targets through tailored rewards and asymmetric advantage normalization: model-generated reasoning can receive negative advantages, whereas retrieved factual tokens receive only non-negative updates. These mechanisms ensure the model internalizes valid knowledge without allowing premature external targets to distract from mastering reliable search strategies.

Extensive experiments on Qwen2.5-3B and Olmo-3-7B across seven single-hop and multi-hop QA benchmarks demonstrate the effectiveness of LfSearch. While naively unmasking tool responses typically causes severe performance degradation, our framework successfully circumvents this optimization collapse. Empowered by the learned knowledge, LfSearch consistently outperforms prompt-based and RL-tuned baselines, demonstrating that internalized facts directly translate to stronger dynamic search performance. Finally, targeted evaluations on knowledge-retention sets explicitly confirm that our framework effectively distills retrieved facts into parametric memory, enabling the model to learn from its environment. In summary, our main contributions are threefold:

- We identify the retrieved-token masking dilemma in search-augmented LLMs: permanent masking prevents knowledge internalization, while indiscriminate unmasking destabilizes optimization.
- We propose LfSearch, a dual-objective RL framework that shifts the paradigm from merely learning to search toward searching to learn, featuring an adaptive masking curriculum and asymmetric advantage normalization to safely internalize retrieved facts without disrupting tool-use learning.
- We empirically demonstrate across seven QA benchmarks that LfSearch circumvents the performance collapse of naive unmasking, translating internalized knowledge into superior dynamic search performance while successfully distilling external knowledge into parametric memory.

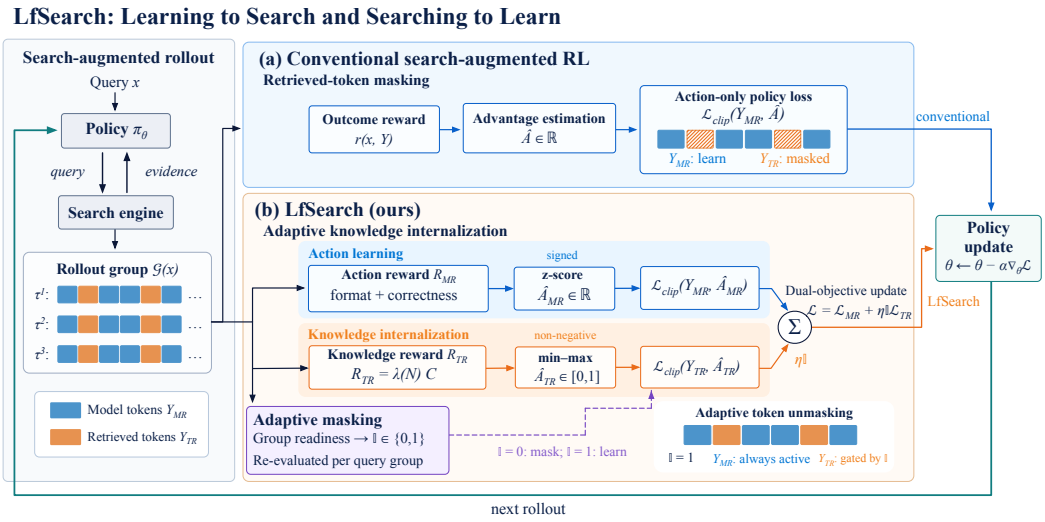

Figure 2: Overview of LfSearch. Conventional search-augmented RL masks retrieved tokens and optimizes only model-generated actions, whereas LfSearch adaptively activates a knowledge-internalization objective and combines it with action learning through a dual-objective update.

2 METHODOLOGY

In this section, we present LfSearch, an end-to-end group-relative reinforcement learning framework that shifts the paradigm of search-augmented LLMs from merely learning to search toward searching to learn. Figure 2 presents the overall training pipeline, contrasting conventional action-only optimization with the adaptive dual-objective update of LfSearch. LfSearch applies a unified dual-objective policy gradient to model-generated actions and externally retrieved knowledge. To internalize retrieved knowledge without disrupting the acquisition of reliable tool-use behavior, we introduce an adaptive masking curriculum governed by the readiness of each rollout group. Finally, we use phase-dependent rewards and asymmetric advantage normalization to provide distinct learning signals for tool use and knowledge internalization. Implementation details are provided in Appendix A.

2.1 PRELIMINARIES: RL WITH SEARCH ENGINES

We formulate the LLM's reasoning process interacting with a search engine $\mathcal{R}$ as a generative trajectory Y. In standard reinforcement learning frameworks for LLMs (Ouyang et al., 2022), the objective function is formulated to maximize the expected reward while maintaining a KL-divergence constraint, as follows:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, Y\sim\pi_\theta(\cdot|x;\mathcal{R})}[r(x, Y)] - \beta D_{KL}[\pi_\theta(Y|x;\mathcal{R})||\pi_{ref}(Y|x;\mathcal{R})] \tag{1}$$

In this formulation, π_θ is the policy LLM being trained, π_{ref} is the reference LLM, $r(x, Y)$ denotes the reward function, and D_{KL} represents the KL-divergence measure. The variable x denotes input samples drawn from the dataset $\mathcal{D}$, and Y represents the generated outputs interleaved with the search engine calling results. Crucially, the term $\pi_\theta(\cdot|x;\mathcal{R})$ denotes the interleaved retrieval-and-reasoning policy, effectively representing the joint process of the model's generation and the external search engine's retrieval.

To ensure optimization stability during training, recent frameworks (Jin et al., 2025) apply a strict loss masking mechanism to the retrieved tokens. Because applying the same optimization to retrieved tokens can lead to unintended learning dynamics, the policy gradient ∇_θ is computed exclusively over the tokens generated by the model, excluding the retrieved content from the optimization process. While this prevents the model from diverging, it fundamentally bottlenecks its cognitive capacity. The external knowledge remains strictly as read-only context, and the factual knowledge itself is never distilled into the parametric weights θ.

2.2 UNIFIED DUAL-OBJECTIVE POLICY GRADIENT

To formulate our learning objective, we first decouple the components of a search-augmented trajectory Y. We partition the tokens into two mutually exclusive sets based on their origin:

- **Model Response (Y_{MR}):** Tokens autonomously generated by the policy π_θ (e.g., `<think>` reasoning chains, `<tool_call>` queries, and final answers).
- **Tool Response (Y_{TR}):** Text tokens returned by the external environment $\mathcal{R}$ (e.g., retrieved passages wrapped in `<tool_response>`).

In conventional search-augmented RL frameworks, a clipped policy-gradient objective is applied exclusively to the model's own generations, while group-relative normalization supplies the trajectory-level advantages (Schulman et al., 2017; Shao et al., 2024). Specifically, the learning objective isolates Y_{MR} as the sole optimization target while completely masking out the tool responses:

$$\mathcal{L}_{\text{clip}}(Y_{MR}, \hat{A}_{MR}) = -\mathbb{E}\left[\sum_{y_t \in Y_{MR}} \min\left(\rho_t \hat{A}_{MR}, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\hat{A}_{MR}\right)\right] \tag{2}$$

where $\rho_t = \frac{\pi_\theta(y_t|y_{<t},x,\mathcal{R})}{\pi_{old}(y_t|y_{<t},x,\mathcal{R})}$, and $\hat{A}_{MR}$ is the estimated trajectory-level advantage broadcast to all model-generated tokens. Under this paradigm, the external knowledge Y_{TR} acts strictly as a read-only condition.

To overcome this representation bottleneck, LfSearch introduces a unified dual-objective policy gradient. Instead of confining the learning variables to Y_{MR}, we explicitly unlock Y_{TR} and integrate both sets into a shared optimization space. We define our total loss as:

$$\mathcal{L}_{\text{LfSearch}}(\theta) = \mathcal{L}_{\text{clip}}(Y_{MR}, \hat{A}_{MR}) + \eta \cdot \mathbb{I}_{unmask} \cdot \mathcal{L}_{\text{clip}}(Y_{TR}, \hat{A}_{TR}) \tag{3}$$

where η is a scaling coefficient balancing the two branches, and $\mathbb{I}_{unmask} \in \{0, 1\}$ is a dynamic indicator controlling the activation of the knowledge internalization objective. Both branches use the same PPO-style clipped surrogate but receive separately constructed masks and group-relative advantages. When activated, the tool-response branch maximizes the likelihood of the retrieved facts under the model's policy distribution. This mechanism effectively distills the non-parametric search results directly into the model's internal memory.

2.3 ADAPTIVE CURRICULUM VIA DYNAMIC MASKING

The indicator $\mathbb{I}_{unmask}$ determines whether the Y_{TR} branch is activated for a query in the current update. Prematurely activating this branch would expose the model to irrelevant evidence retrieved via poorly formulated early queries. We therefore evaluate a query-level rollout group before unmasking its retrieved content. For trajectory i, let $F_i = \min(1, \sum_t r^{(i,t)}_{format})$ denote its accumulated format score and let $C_i \in \{0, 1\}$ indicate a correct final answer. We first compute a readiness score:

$$S_i = -1 + F_i + C_i \tag{4}$$

Crucially, the decision to unlock the tool-response branch is made at the group level for each specific query. It is enabled only when the entire sampled rollout group $\mathcal{G}(x)$ satisfies the readiness threshold τ_{ready} and contains at least one correct answer. We formalize this activation condition as:

$$\mathbb{I}_{unmask}(x) = \begin{cases} 1, & \text{if } \min_{i \in \mathcal{G}(x)} S_i \geq \tau_{ready} \ \wedge \ \max_{i \in \mathcal{G}(x)} C_i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

When $\mathbb{I}_{unmask} = 0$, retrieved tokens remain masked and training focuses on model-generated actions. When $\mathbb{I}_{unmask} = 1$, the tool-response branch and its corresponding phase-dependent reward mechanism are activated for that group. By dynamically evaluating readiness on a per-query basis, the framework naturally forms a curriculum (Bengio et al., 2009): it progressively shifts the training focus from acquiring basic search skills to internalizing external knowledge as the policy's proficiency improves.

2.4 TAILORED REWARD AND NORMALIZATION MECHANISMS

Although Y_{MR} and Y_{TR} share the same clipped policy-loss architecture, they represent fundamentally different cognitive processes: mastering tool usage versus internalizing factual knowledge. To accurately reflect this divergence, we design tailored reward and normalization mechanisms that specifically adapt to their distinct optimization goals. We first define how raw rewards are distinctly assigned to the model's actions and the tool's responses, and subsequently detail how these rewards are normalized into advantages.

2.4.1 OBJECTIVE-ADAPTIVE REWARD ASSIGNMENT

We introduce the reward assignment strategies in an order that mirrors the agent's cognitive curriculum: first mastering how to search, and subsequently internalizing what to learn.

Learning to Search The reward formulation for model-generated actions integrates a format-adherence score with a dynamic correctness objective. Once the adaptive curriculum is triggered, the correctness signal is weighted by a step-dependent efficiency factor:

$$R_{MR}^{(i)} = -1 + F_i + \left[(1 - \mathbb{I}_{unmask}) + \mathbb{I}_{unmask}\lambda(N_i)\right] C_i, \tag{6}$$

where N_i is the number of interaction steps and $\lambda(N_i)$ assigns larger correctness rewards to shorter successful trajectories. Before activation, the model-response branch retains the base penalty, format reward, and an unweighted correctness reward. After activation, the correctness term becomes efficiency-aware. This not only penalizes redundant exploration but also explicitly incentivizes the model to leverage its newly internalized parametric knowledge, encouraging it to solve problems with minimal external search invocations.

Searching to Learn To assess the value of the external knowledge, the tool response directly inherits the trajectory's efficiency-aware correctness score:

$$R_{TR}^{(i)} = \mathbb{I}_{unmask}\lambda(N_i)C_i. \tag{7}$$

Therefore, the tool-response reward is zero before activation and isolates the factual utility by excluding the base penalty and format score. The underlying intuition is deeply tied to the signal-to-noise ratio of the retrieved text. External search inevitably returns a mixture of relevant facts and irrelevant noise. A correct final answer confirms that the retrieved content indeed contains task-relevant information. Furthermore, a shorter interaction trajectory implies that this relevant knowledge accounts for a larger proportion of the total retrieved text. By inheriting the efficiency weight $\lambda(N_i)$, the tool-response branch applies a stronger learning signal to trajectories with higher knowledge density, effectively amplifying the internalization of high-quality evidence.

2.4.2 ASYMMETRIC ADVANTAGE NORMALIZATION

Once raw rewards are assigned, they must be converted into advantages to drive the policy gradient. This step embodies a core philosophical distinction: model-generated reasoning can be flawed, whereas retrieved factual knowledge is inherently valid and should never be penalized.

Learning to Search Since the model's autonomous actions can be suboptimal, excessively reliant on search, or hallucinatory, the advantage must be capable of suppressing bad behaviors. We apply GRPO-style group-relative z-score normalization across the trajectories sampled for the same query (Shao et al., 2024; Guo et al., 2025):

$$\hat{A}_{MR}^{(i)} = \frac{R_{MR}^{(i)} - \mu(R_{MR})}{\sigma(R_{MR})} \tag{8}$$

where μ and σ denote the mean and standard deviation computed over the group $\mathcal{G}(x)$. The resulting trajectory-level advantage is broadcast to its model-generated tokens. This allows $\hat{A}_{MR} \in \mathbb{R}$ to be negative, effectively pushing the policy away from incorrect reasoning paths.

Table 1: EM (%) across single-hop and multi-hop QA benchmarks based on Qwen2.5-3B-Instruct.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Direct Inference	10.97	6.23	9.97	15.35	24.17	1.86	4.00	10.36
CoT	10.14	35.07	11.38	15.18	17.62	3.02	16.00	15.49
RAG	32.71	53.33	37.93	24.00	15.82	5.17	16.80	26.54
IRCoT	33.43	53.54	36.68	25.62	21.29	4.96	17.60	27.59
Search-o1	24.18	42.31	24.86	19.93	24.14	5.71	26.40	23.93
Search-R1	48.22	62.98	42.76	39.80	46.89	16.71	28.80	40.88
Search-R1 w.o. mask	39.97	57.87	36.68	30.97	40.39	9.21	22.37	33.92
LfSearch	**49.70**	**63.02**	**43.95**	**39.95**	**47.72**	**17.13**	**34.40**	**42.27**

Table 2: EM (%) across single-hop and multi-hop QA benchmarks based on Olmo-3-7B-Instruct.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Direct Inference	11.02	33.12	12.55	15.76	28.27	3.23	5.60	15.65
CoT	11.27	33.96	12.62	16.79	26.77	3.31	13.60	16.90
RAG	27.92	48.34	34.42	20.49	11.44	3.56	15.20	23.05
IRCoT	26.65	48.46	37.02	19.76	8.64	5.34	17.60	23.35
Search-o1	19.36	43.24	27.09	21.26	20.13	5.50	17.60	22.03
Search-R1	47.56	61.53	41.86	33.84	33.91	13.45	28.00	37.16
Search-R1 w.o. mask	38.45	56.78	35.93	26.41	27.48	8.53	22.70	30.90
LfSearch	**47.89**	**63.42**	**43.86**	**34.06**	**34.57**	**14.11**	**33.60**	**38.79**

Searching to Learn Conversely, objective knowledge returned by the environment is inherently valid. It must not be penalized with a negative advantage, even if alternative trajectories in the same group achieve higher scores and, consequently, higher knowledge density. To enforce this, we apply a strict min-max normalization:

$$\hat{A}_{TR}^{(i)} = \frac{R_{TR}^{(i)} - \min_j R_{TR}^{(j)}}{\max_j R_{TR}^{(j)} - \min_j R_{TR}^{(j)}} \tag{9}$$

where the minimum and maximum are similarly taken over the query group $\mathcal{G}(x)$. This mathematically bounds $\hat{A}_{TR} \in [0, 1]$. Under this constraint, the objective solely maximizes the likelihood of predicting factual knowledge, scaling the update magnitude based on knowledge density while ensuring the parametric weights do not diverge from valid external information.

3 Experiments

This section presents a comprehensive evaluation of LfSearch across diverse single-hop and multi-hop question answering benchmarks. We begin by detailing the experimental setup and then analyze the main results, illustrating how the proposed framework circumvents the masking dilemma to deliver superior dynamic search capabilities. Subsequently, we conduct targeted evaluations to verify parametric knowledge internalization. We conclude with ablation studies to investigate the impact of the core components within our dual-objective framework.

3.1 Experiment Settings

We evaluate LfSearch using Qwen2.5-3B-Instruct (Yang et al., 2024) and Olmo-3-7B-Instruct (Team Olmo, 2025) across seven diverse single-hop and multi-hop QA benchmarks. We compare our framework against prompt-based pure inference, iterative search methods (e.g., IRCoT (Trivedi et al., 2023), Search-o1 (Li et al., 2025)), and RL-tuned search baselines, specifically Search-R1 (Jin et al., 2025). Crucially, to explicitly demonstrate the optimization collapse, we compare the standard masked Search-R1 with a naively unmasked variant (Search-R1 w.o. mask). Across all datasets, we uniformly adopt Exact Match (EM) as the primary evaluation metric. Complete details regarding training datasets, benchmark categorization, comprehensive baseline definitions, and metric formulations are provided in Appendix B.

Table 3: EM (%) on the Paraphrase set and EvidenceQA set with and without search.

Backbone	Method	No retrieval			Search enabled		
		Paraphrase	EvidenceQA	Avg.	Paraphrase	EvidenceQA	Avg.
Qwen2.5-3B-Instruct	Base	33.50	18.75	26.13	56.50	62.25	59.38
	Search-R1	28.75	17.25	23.00	84.00	92.25	88.13
	LfSearch	**36.00**	**21.25**	**28.63**	**87.25**	**94.25**	**90.75**
Olmo-3-7B-Instruct	Base	35.25	19.50	27.38	25.50	17.75	21.63
	Search-R1	35.75	17.25	26.50	80.25	89.50	84.88
	LfSearch	**37.75**	**20.25**	**29.00**	**82.25**	**91.50**	**86.88**

3.2 Main Results: Circumventing the Masking Dilemma

Tables 1 and 2 present the comprehensive QA performance of all evaluated methods. LfSearch consistently delivers the highest average EM score across both backbones, significantly outperforming all prompt-based and RL-tuned baselines on both single-hop and multi-hop datasets.

Overcoming Optimization Collapse A pivotal observation from our results is the stark contrast between the two Search-R1 variants, which perfectly illustrates the retrieved-token masking dilemma introduced in Section 1. When tool responses are naively unmasked (Search-R1 w.o. mask), performance severely degrades relative to the standard masked baseline (Search-R1)—dropping average EM by 6.96 points on Qwen2.5-3B-Instruct and 6.26 points on Olmo-3-7B-Instruct. This confirms that indiscriminate optimization over external text introduces fatal noise that disrupts the policy optimization. In contrast, LfSearch safely absorbs valid facts through its adaptive masking curriculum, circumventing this collapse and achieving a substantial +8.35 and +7.89 average EM improvement over the naively unmasked variant on the two models, respectively.

Superior Dynamic Search for Complex Reasoning The performance gains of LfSearch are particularly pronounced on challenging multi-hop benchmarks. For instance, on Bamboogle, LfSearch achieves an absolute improvement of +5.6 EM over the strongest baseline on both architectures. Prompt-based iterative methods frequently fail to trace complex reasoning paths, while the standard masked RL baseline is bottlenecked by treating retrieval purely as read-only context. By successfully internalizing knowledge during training, LfSearch overcomes this bottleneck and consistently outperforms the search-only baseline. The superior performance indicates that knowledge internalization does not compromise tool-use ability. Rather, the enriched memory serves as a strong foundation for solving complex, multi-step queries.

3.3 Knowledge Internalization Evaluation

To explicitly verify whether the external knowledge retrieved during training is distilled into parametric memory rather than merely accessed as read-only context, we construct two 400-example evaluation sets derived from the NQ and HotpotQA training instances. The Paraphrase set consists of rephrased versions of the original training questions, whereas the EvidenceQA set introduces newly generated questions grounded in the corresponding retrieved passages. We report EM scores under both no-retrieval and search-enabled settings.

Preventing Knowledge Degradation Table 3 jointly evaluates parametric knowledge retention and dynamic search performance. A crucial finding is that Search-R1 suffers from the degradation of its original parametric knowledge. On Qwen2.5-3B-Instruct, its no-retrieval average drops from the Base model's 26.13% down to 23.00%. This indicates that optimizing exclusively for search execution may erode the model's pre-existing memory. In contrast, LfSearch significantly boosts no-retrieval accuracy to 28.63%. This verifies that the dual-objective policy gradient effectively forces the model to memorize and internalize the external facts encountered during training. Furthermore, because this pattern holds strongly on the newly generated EvidenceQA set, the gains originate from genuine knowledge distillation rather than surface-form memorization of the original prompts.

Table 4: Downstream QA EM (%) ablation on Qwen2.5-3B-Instruct.

Method	η	Single-Hop QA	Multi-Hop QA		Avg.
		NQ	HotpotQA	Bamboogle	
Search-R1	–	48.22	39.80	28.80	38.94
LfSearch	0.5	48.95	39.84	33.60	40.80
	1.0	**49.70**	**39.95**	**34.40**	**41.35**
	2.0	48.03	39.42	32.00	39.82

Table 5: Knowledge-retention EM (%) ablation on Qwen2.5-3B-Instruct.

Method	η	No retrieval			Search enabled		
		Paraphrase	EvidenceQA	Avg.	Paraphrase	EvidenceQA	Avg.
Base	–	33.50	18.75	26.13	56.50	62.25	59.38
Search-R1	–	28.75	17.25	23.00	84.00	92.25	88.13
LfSearch	0.5	35.25	17.75	26.50	81.75	**94.75**	88.25
	1.0	36.00	**21.25**	**28.63**	**87.25**	94.25	**90.75**
	2.0	**37.50**	19.50	28.50	82.50	92.75	87.63

Self-Enhancement in Search-Enabled Scenarios More importantly, these parametric gains do not come at the expense of tool-use proficiency. When retrieval is enabled, LfSearch remains the most performant method across the board. This validates a core hypothesis of our work: parametric memory and external search are mutually reinforcing. Instead of treating acquired facts as a static fallback when search is unavailable, the internalized knowledge acts as a strong cognitive prior that synergizes with real-time tool calls, enabling the model to navigate search spaces more accurately.

3.4 Ablation Studies

To investigate the learning dynamics of our dual-objective framework, we ablate the coefficient η from Eq. (3), which controls the weight of the knowledge internalization branch. Tables 4 and 5 illustrate the non-monotonic impact of this parameter on Qwen2.5-3B-Instruct.

The optimal setting is not obtained by simply maximizing the weight on retrieved facts. Instead, a balanced coefficient ($\eta = 1.0$) yields the best overall trade-off. When the internalization weight is too small ($\eta = 0.5$), the no-retrieval performance remains sub-optimal, indicating that the model still heavily relies on real-time retrieval and fails to deeply internalize the evidence. Conversely, an excessively large weight ($\eta = 2.0$) heavily prioritizes parametric memorization—achieving the highest no-retrieval Paraphrase score—but simultaneously degrades search-enabled averages and downstream QA generalization. This degradation suggests that over-emphasizing fact reconstruction distracts the model from mastering tool-use operations, breaking the delicate balance of the cognitive curriculum. The default choice $\eta = 1.0$ serves as a robust compromise. This ablation highlights a fundamental design principle for search-augmented RL: the learning signals for tool-use execution and knowledge internalization must be explicitly decoupled and carefully balanced. This balance ensures that the model leverages external facts to augment rather than replace its dynamic problem-solving capacity.

4 Related Work

4.1 Reinforcement Learning for LLM Reasoning and Tool Use

Reinforcement learning has become a central paradigm for post-training LLMs beyond supervised imitation. Early alignment pipelines commonly adopt proximal policy optimization (PPO) to optimize preference-based rewards while constraining the updated policy to remain close to a reference model (Ouyang et al., 2022; Schulman et al., 2017). More recent work applies outcome-based or rule-based rewards to reasoning tasks, enabling models to acquire long-form reasoning behaviors through online exploration (Shao et al., 2024; Guo et al., 2025; Du et al., 2025; Xie et al., 2025). In particular, group-relative optimization estimates advantages from multiple responses without re-

quiring a separate critic (Shao et al., 2024). Concurrently, scalable variants have been proposed to improve the stability and efficiency of long-horizon reasoning training (Yu et al., 2025; Yuan et al., 2025). REINFORCE-style methods similarly simplify policy optimization and provide competitive alternatives to conventional PPO training (Ahmadian et al., 2024; Li et al., 2023; Hu, 2025).

RL has also expanded from closed-book reasoning to interactive settings in which an LLM must decide when and how to invoke external tools (Cheng et al., 2025; Feng et al., 2025). Compared with ordinary text generation, these settings produce trajectories that interleave policy-generated reasoning and actions with observations returned by an environment. Search-R1 applies outcome-driven RL to such trajectories, allowing a model to learn multi-turn search strategies from final-answer supervision (Jin et al., 2025). However, to prevent optimization collapse, these existing frameworks completely mask out the environment's responses, treating retrieved observations strictly as read-only context. LfSearch builds on this algorithmic foundation but introduces an adaptive masking curriculum and asymmetric advantages for retrieved content. This extends RL from merely learning tool-use behavior to genuinely internalizing the knowledge acquired through interaction.

4.2 Search-Augmented LLMs and Agentic Retrieval

Despite advances in reasoning, LLMs remain limited on knowledge-intensive tasks by static parametric memory and hallucinations (OpenAI, 2024; Guo et al., 2025; Du et al., 2025; Qwen Team, 2025). Retrieval-augmented generation (RAG) (Lewis et al., 2020) conditions generation on passages returned by a single retrieval step. While simple and effective for single-hop questions, this pipeline is inherently rigid: it cannot revise queries or perform multi-turn retrieval when intermediate reasoning reveals missing information. Iterative and multi-hop RAG methods reduce this rigidity by alternating retrieval and generation or by self-critiquing retrieved evidence (Asai et al., 2024; Jiang et al., 2023; Shao et al., 2023). However, they still rely on fixed heuristics or prompting rather than learning a dynamic search policy.

Search-as-a-tool methods instead expose the search engine as an interactive tool that the model can invoke autonomously during reasoning. Prompt-based approaches such as ReAct (Yao et al., 2023) and IRCoT (Trivedi et al., 2023) interleave thought and retrieval through prompting, while tool-use training teaches models to call external APIs or search engines (Schick et al., 2023; Nakano et al., 2021). More recent work directly optimizes search behavior with RL, training models to generate multi-turn trajectories that combine internal reasoning with explicit search calls (Chen et al., 2025; Wang et al., 2025b;a). These agents learn when and how to retrieve more flexibly than static RAG or prompting baselines. Nevertheless, retrieved evidence is generally utilized only as transient context and completely excluded from policy optimization. LfSearch complements this line of work by enabling the model to internalize useful retrieved facts into its parameters while retaining its robust ability to conduct dynamic search.

5 Conclusion

In this work, we identify the retrieved-token masking dilemma within reinforcement learning frameworks for search-augmented reasoning: permanent masking stabilizes training but prevents knowledge internalization, whereas naive unmasking of retrieved tokens precipitates optimization collapse. To resolve this tension, we propose LfSearch, a unified dual-objective policy gradient framework that transitions the learning paradigm from merely learning to search toward searching to learn. The framework introduces an adaptive masking curriculum that unlocks retrieved-token optimization only after the model acquires stable tool-use proficiency, coupled with target-specific reward formulation and asymmetric advantage normalization. These mechanisms explicitly decouple tool-use execution from knowledge internalization, allowing the policy to safely absorb high-quality external facts while protecting parametric weights from the noise inherent in early-stage exploration. Extensive empirical evaluations across diverse single-hop and multi-hop question answering benchmarks demonstrate that LfSearch consistently outperforms standard prompt-based and RL-tuned baselines. Dedicated knowledge-retention analyses further confirm that the framework successfully distills external evidence into parametric memory. Ultimately, we show that this internalized knowledge does not compromise tool-use proficiency. Rather, it acts as a strong cognitive prior that synergizes with dynamic search, establishing a robust foundation for complex reasoning tasks.

AI USE STATEMENT

Generative AI tools were used only to edit the manuscript text and assist with LaTeX formatting, including migration to the ICLR 2027 template. They were not used to develop research ideas, implement research code, conduct experiments, or analyze results. The authors take responsibility for the final content of the manuscript.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Fan Yang, Zenan Zhou, Weipeng Chen, Haofen Wang, Jeff Z. Pan, Wen Zhang, and Huajun Chen. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. *arXiv preprint arXiv:2503.19470*, 2025.

Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruiran Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. Agent-R1: Training Powerful LLM Agents with End-to-End Reinforcement Learning. *arXiv preprint arXiv:2511.14460*, 2025.

Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi L1.5: Scaling Reinforcement Learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. *arXiv preprint arXiv:2505.10978*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Jian Hu. REINFORCE++: A Simple and Efficient Approach for Aligning Large Language Models. *arXiv preprint arXiv:2501.03262*, 2025.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Ö. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic Search-Enhanced Large Reasoning Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Alex Mallen, Akari Asai, Victor Zhong, Nazneen Rajani, Patrick Lewis, Luke Zettlemoyer, Pang Wei Koh, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Introducing OpenAI o1. https://openai.com/o1/, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Qwen Team. Qwen3. https://github.com/QwenLM/Qwen3/blob/main/Qwen3_Technical_Report.pdf, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL https://arxiv.org/abs/2512.13961.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 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)*, pp. 10014–10037, 2023.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn LLM Agents. *arXiv preprint arXiv:2510.14967*, 2025a.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. *arXiv preprint arXiv:2505.15107*, 2025b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-RL: Unleashing LLM Reasoning with Rule-Based Reinforcement Learning. *arXiv preprint arXiv:2502.14768*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. VAPO: Efficient and Reliable Reinforcement Learning for Advanced Reasoning Tasks. *arXiv preprint arXiv:2504.05118*, 2025.

A Detailed Methodological Implementation

A.1 Trajectory Serialization and Token Masks

During each interaction step, the trajectory records the model-generated action. If a search is invoked, the complete tool observation is appended to the same response sequence. The appended observation includes the chat-role delimiters and the opening and closing <tool_response> tags surrounding the raw retrieved content.

To facilitate our dual-objective framework, we maintain separate position-wise masks. The model-response mask selects all tokens generated by the policy. The tool-response mask is activated exclusively for unmasked queries and selects only the raw retrieved content. All model-generated tokens, chat-role delimiters, wrapper tags, and padding tokens remain masked. To prevent external text from interfering with reward evaluation, the reward computation function parses only the model-generated segments of each step, ensuring that appended observations do not affect format validation or answer extraction. If the complete tool observation exceeds the remaining response budget, the trajectory terminates immediately after that tool call without appending a truncated observation.

A.2 Curriculum and Reward Instantiation

Let $V_{i,s}^{tool} \in \{0,1\}$ indicate that step s constitutes a complete and parseable <think>–<tool_call> action, and let $V_{i,s}^{ans} \in \{0,1\}$ indicate that it strictly adheres to the <think>–<answer> final-answer format. The per-step format reward is defined as follows:

$$f_{i,s} = \begin{cases} 0.5, & V_{i,s}^{tool} = 1, \\ 0.6, & V_{i,s}^{ans} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The accumulated format score is capped at 1.0:

$$F_i = \min\left(1, \sum_s f_{i,s}\right). \tag{11}$$

Let $\widehat{a}_i$ be the answer extracted from the final step and $\mathcal{A}_i$ be the set of acceptable ground-truth answers. The correctness indicator aligns with the SubEM evaluation metric:

$$C_i = \mathbb{I}[\text{ValidFinal}(i) \ \wedge \ \exists a \in \mathcal{A}_i : \text{norm}(a) \subseteq \text{norm}(\widehat{a}_i)], \tag{12}$$

where the $\text{norm}(\cdot)$ function lowercases the text, removes punctuation and articles, and collapses whitespace. The pre-activation trajectory reward is formulated as:

$$R_i^{(1)} = -1 + F_i + C_i. \tag{13}$$

For each query x, we sample a group $\mathcal{G}(x)$ consisting of five trajectories. The adaptive masking curriculum activates the tool-response branch according to the following condition:

$$\mathbb{I}_{unmask}(x) = \mathbb{I}\left[\min_{i \in \mathcal{G}(x)} R_i^{(1)} \geq -0.4 \ \wedge \ \max_{i \in \mathcal{G}(x)} R_i^{(1)} \geq 0.6\right]. \tag{14}$$

Based on the defined format and correctness bounds, the minimum condition ensures that all trajectories in the group exhibit basic formatting proficiency, while the maximum condition guarantees that at least one trajectory successfully solves the problem.

To incentivize efficient tool use and penalize redundant exploration, the step-dependent efficiency weight $\lambda(N)$ monotonically decreases with the total number of interaction steps N:

$$\lambda(N) = \begin{cases} 2.5, & N = 1, \\ 1.75, & N = 2, \\ 1.5, & N = 3, \\ 1.25, & N = 4, \\ 1.0, & N \geq 5. \end{cases} \tag{15}$$

Ultimately, the finalized rewards utilized by the two optimization branches are instantiated as:

$$R_{MR}^{(i)} = -1 + F_i + [(1 - \mathbb{I}_{unmask}) + \mathbb{I}_{unmask}\lambda(N_i)]\, C_i, \tag{16}$$

$$R_{TR}^{(i)} = \mathbb{I}_{unmask}\lambda(N_i) C_i. \tag{17}$$

A.3 Tool-Response Advantage Construction

For an activated query, let $\mathcal{T}(x)$ denote the subset of trajectories within the group that contain at least one retrieved-content token. We use $q_{\min}$ and $q_{\max}$ to denote the minimum and maximum tool-response rewards over this subset. When $\mathcal{T}(x)$ is empty, both values default to zero. To compute the trajectory-level tool-response advantage, we apply strict min-max normalization:

$$\bar{A}_{TR}^{(i)} = \begin{cases} 0, & q_{\max} = 0, \\ 1, & q_{\max} = q_{\min} > 0, \\ \dfrac{R_{TR}^{(i)} - q_{\min}}{q_{\max} - q_{\min}}, & q_{\max} > q_{\min}. \end{cases} \tag{18}$$

This trajectory-level advantage is subsequently broadcast to all retrieved-content token positions selected by the tool-response mask. To ensure numerical stability, this mask is applied only when $q_{\max} > 0$. Otherwise, the mask remains zero and the secondary loss is skipped. Furthermore, in the positive constant case ($q_{\max} = q_{\min} > 0$), the advantage is explicitly clipped to 1 to prevent an undefined zero denominator while maintaining a positive update signal.

B Additional Experiment Details

B.1 Datasets and Evaluation Metrics

We construct the training set by sampling from HotpotQA (Yang et al., 2018) and Natural Questions (NQ) (Kwiatkowski et al., 2019), balancing multi-hop and single-hop reasoning questions. The comprehensive evaluation suite spans single-hop tasks (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023)) alongside multi-hop tasks (HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023)). Across all datasets, we uniformly adopt Exact Match (EM) as the primary evaluation metric. Specifically, a generated prediction is considered correct if its normalized text contains at least one of the normalized ground-truth answers.

B.2 Baseline Configurations

To evaluate the effectiveness of LfSearch, we compare it against three categories of competitive baselines. The first category consists of prompt-based pure inference methods, including Naive Direct Inference, Chain-of-Thought (CoT) (Wei et al., 2022), and standard RAG (Lewis et al., 2020). The second category encompasses prompt-based iterative inference methods, including IR-CoT (Trivedi et al., 2023) and Search-o1 (Li et al., 2025). The third category comprises RL-tuning-based methods, specifically Search-R1 (Jin et al., 2025). Crucially, to explicitly demonstrate the optimization collapse caused by noisy external text, we evaluate the standard Search-R1 (where tool responses are permanently masked) against Search-R1 w.o. mask (where tool responses are indiscriminately optimized from the start).

B.3 Optimization Configuration

Our main experiments utilize E5-base-v2 (Wang et al., 2022) to retrieve the top three passages from the 2018 Wikipedia dump. For each training query, we sample five trajectories using a decoding temperature of 1.0 and a top-p value of 0.95, permitting a maximum of five interaction steps per rollout. Both the maximum prompt length and the maximum response length are configured to 8192 tokens.

We train the model for one epoch using the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 1×10^{-6} and a weight decay of 0.01. The training configuration employs a global batch size of 64. Unless otherwise specified, the tool-response loss coefficient is set to $\eta = 1.0$. Alternative values are analyzed in Section 3.4.

C LIMITATIONS

While LfSearch successfully bridges tool-use optimization and knowledge internalization, we acknowledge several limitations in the current framework. First, the current objective uniformly distributes the internalization reward across all retrieved tokens within a successful trajectory. Implementing span-level credit assignment would enable the model to more precisely isolate vital facts from noisy or partially relevant context. Second, our empirical validation is primarily conducted on English Wikipedia-based open-domain QA using foundation models up to the 7B parameter scale. The framework's robustness under heterogeneous search APIs, highly noisy dynamic web environments, and massive-scale foundation models remains to be fully explored. Finally, the dual-objective formulation introduces the hyperparameter η to explicitly balance the policy gradient signals. Although empirical results demonstrate robustness around the default configuration, investigating curriculum-driven scheduling for this coefficient presents a promising direction for future research.