Eval-RAR: Evaluation-Driven Retrieval-Augmented Reasoning via Reinforcement Learning

Heng Yu[1], **Rui Li**[1], **Qi Liu**[1,2*], **Wenjun Feng**[1], **Junfeng Kang**[1], **Yi Zhan**[1]
[1]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{yh112358_1321,ruili2000}@mail.ustc.edu.cn, qiliuql@ustc.edu.cn
{fengwenjun,kangjf,zy0119}@mail.ustc.edu.cn

Abstract

Retrieval-augmented generation (RAG) effectively extends the knowledge boundaries of large language models (LLMs) for complex tasks, yet current paradigms typically optimize for an interleaving of reasoning and retrieval, where models fail to critically evaluate retrieved information against the target question. Most existing methods rely on sparse outcome-based rewards, failing to provide explicit supervision for the internal reasoning process or to diagnose information inadequacy. To address this, we propose **Eval-RAR**, an **Eval**uation-driven **R**etrieval-**A**ugmented **R**easoning framework. Eval-RAR introduces a "Search-then-Evaluate" paradigm where the model performs explicit self-evaluation after each search step, generating a rationale to either identify sufficient evidence or specify missing information to guide subsequent queries. To optimize this process, we employ reinforcement learning with a fine-grained evaluation reward, providing intermediate feedback that encourages the model to track core entities and maintain logical consistency. Experiments on seven single-hop and multi-hop QA benchmarks demonstrate that Eval-RAR outperforms existing methods.

1 Introduction

Large language models have demonstrated certain reasoning and decision-making abilities in solving a wide range of complex Natural Language Processing (NLP) tasks (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Liu et al., 2021; Zhuang et al., 2022; Gao et al., 2025). Retrieval-Augmented Generation (RAG) further expands their knowledge boundaries, enabling them to tackle more complicated reasoning challenges (Lewis et al., 2020; Guu et al., 2020; Gao et al., 2024). Recently, by allowing LLMs to engage in multi-step interactions with search engines, systems can autonomously integrate retrieval and reasoning to solve complex problems (Trivedi et al., 2023; Li et al., 2025b). However, the prevailing Retrieval-Augmented Reasoning (RAR) paradigm suffers from a fundamental structural flaw—the "blind trajectory problem." Most existing frameworks optimized via reinforcement learning (RL) operate by optimizing the model's linear process of alternating between reasoning and retrieval actions (Jin et al., 2025; Song et al., 2025; Chen et al., 2025; Zheng et al., 2025b; Zhang et al., 2025a; Wei et al., 2025), we refer to this paradigm as RL-based RAR. This sequential accumulation mechanism often leads to the uncritical absorption of retrieved information, including noisy or irrelevant content, undermining the robustness and accuracy of the reasoning process.

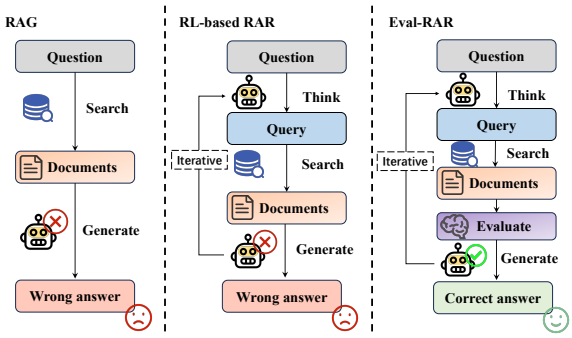


Figure 1: Comparison of RAG, RL-based RAR, and our method, Eval-RAR.

We argue that the primary cause of reasoning failures lies not in insufficient retrieval capability, but rather in the system's lack of accurate diagnostic ability regarding "information inadequacy." In complex multi-step tasks, each retrieval step should be grounded in a precise assessment of the "knowledge gap"—that is, the discrepancy between the currently available knowledge and the requirements of the original query. Without an explicit self-evaluation stage, the model remains in a state of informational blindness: it can neither deter-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 3900–3912
July 2-7, 2026 ©2026 Association for Computational Linguistics

mine whether the existing information is sufficient to answer the question, nor accurately identify what necessary information is still missing. This leads to a vicious cycle of "blind search," where subsequent retrieval queries become repetitive or deviate from the core topic, ultimately causing the reasoning chain to derail or resulting in factual hallucinations.

To break this cycle, we propose Eval-RAR, a framework that introduces a **"Search-then-Evaluate"** paradigm. The key innovation lies in the integration of an explicit self-evaluation step, which serves as the control hub of the reasoning process to guide subsequent actions. During this step, the model must explicitly evaluate the alignment between the retrieved information and the original problem: (1) if the information is sufficient to answer the question, it should specify which retrieved content supports the answer; (2) if the information is insufficient, it should identify the specific information gap, thereby providing clear semantic guidance for subsequent retrieval and reasoning steps. By establishing this "question-centric" checkpoint, we ensure that the reasoning process remains consistently grounded and retrieval actions stay purposefully directed.

However, optimizing such complex behaviors poses a significant challenge to traditional reinforcement learning approaches, as they typically rely on outcome reward mechanisms. The conventional outcome-based reward mechanism only provides rewards for the final correct answer, making it difficult to offer fine-grained incentives for the nuanced capabilities demonstrated during intermediate evaluation steps. To address this, we introduce an **evaluation reward** to align evaluation with the answer. Specifically, when the answer is incorrect, the evaluation reward supplements fine-grained process-based rewards, explicitly encouraging the model to conduct active evaluation and knowledge supplementation based on the original question and existing information.

In summary, our contributions are as follows:

- We propose Eval-RAR, a retrieval-augmented reasoning framework that integrates an explicit self-evaluation step, enabling the model to evaluate information sufficiency and identify knowledge gaps to guide further retrieval and reasoning.
- We design a fine-grained evaluation reward within the RL training process, which provides intermediate feedback on the model's evaluation, effectively alleviating outcome reward sparsity.
- We conduct extensive experiments across seven single-hop and multi-hop QA benchmarks. Our results demonstrate that Eval-RAR outperforms existing methods.

2 Related Work

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) has been widely adopted to mitigate hallucinations and extend the knowledge boundaries of Large Language Models (LLMs) (Lewis et al., 2020; Guu et al., 2020; Gao et al., 2024; Yu et al., 2025; Zhang et al., 2025b). Earlier RAG systems typically employ static retrieval strategies, where a fixed number of documents are retrieved based on a query and integrated into the context for final answer generation. However, this single-round interaction faces significant challenges in complex scenarios, as it struggles to ensure the continuity and integrity of the reasoning chain. While several studies have attempted to address these limitations through supervised fine-tuning (SFT) (Asai et al., 2024; Shi et al., 2024; Xu et al., 2024; Lin et al., 2024; Li et al., 2025a), they are still limited by the one-shot retrieval paradigm, which often fails to capture the complex dependencies required for multi-hop reasoning. To bridge this gap, recent advancements have introduced **retrieval-augmented reasoning**, which integrates real-time retrieval and query generation directly into the step-by-step reasoning process of LLMs (Trivedi et al., 2023; Li et al., 2025b). While this paradigm allows for dynamic interaction with external knowledge, effectively optimizing such a complex process requires sophisticated training strategies beyond traditional supervision.

Reinforcement Learning for LLM Reasoning. The post-training phase of LLMs has recently pivoted toward reinforcement learning (RL) to catalyze advanced reasoning and self-correction capabilities (OpenAI et al., 2024; DeepSeek-AI et al., 2025). While early frameworks established the standard for preference alignment through Proximal Policy Optimization (PPO) (Schulman et al., 2017; Ouyang et al., 2022), the field has shifted toward more resource-efficient strategies like Group Relative Policy Optimization (GRPO) (Shao et al., 2024). By utilizing group-wise advantages, these methods eliminate the need for critic models, sig-

nificantly enhancing performance in logically rigorous domains such as mathematics (Shao et al., 2024) and code generation (DeepSeek-AI et al., 2024). This trend has naturally extended to the RAG landscape, where RL is employed to govern the interplay between internal reasoning and external search (Jin et al., 2025; Song et al., 2025; Chen et al., 2025; Zheng et al., 2025b). For example, Search-R1 (Jin et al., 2025) utilizes outcome-based rewards to incentivize autonomous search behaviors. To move beyond outcome-based rewards, recent works such as AutoRefine (Shi et al., 2025) introduce intermediate rewards by refining retrieved documents after search calls. Furthermore, frameworks like StepSearch (Zheng et al., 2025a) and GlobalRAG (Luo et al., 2025) employ fine-grained rewards to ensure the correctness of intermediate steps. However, these methods typically rely on extensive data augmentation to obtain golden trajectories for training and still lack an explicit evaluation process to identify and correct errors during the reasoning process. In contrast, Eval-RAR introduces a rationale-alignment reward within the RL training pipeline. By anchoring the optimization to explicit evaluation steps that identify evidence or pinpoint missing information, our approach provides a self-correction mechanism that ensures reasoning continuity and factual consistency without strictly depending on golden reasoning paths.

3 Preliminaries

This section defines the task of retrieval-augmented reasoning for question-answering (QA) tasks that require iterative interactions with a search engine.

3.1 Task Formulation

Given an input query q and a search engine that retrieves from a document corpus $\mathcal{C}$, the objective is to generate an accurate answer a through a sequence of reasoning and retrieval actions.

Naive RAG systems typically follow a "one-shot" pipeline. Given a query q, a retriever returns the k most relevant documents $\mathcal{D}_k(q) = \{d_1, d_2, \ldots, d_k\}$. The model then generates the answer a in a single pass: $a \sim P(a|q, \mathcal{D}_k(q))$. While efficient, this approach is often insufficient for queries that require incremental evidence gathering or multi-step logical planning.

In our paradigm, the model treats the task as a multi-step reasoning process. Given q, the LLM generates a reasoning trajectory $\tau = (t_1, t_2, \ldots, t_n)$, where each step t_i represents a discrete reasoning and interaction hop. At each i-th step t_i, the model engages in the following cycle: (1) Based on the current context, the model generates an internal "thought" to determine the next reasoning direction and formulates a specific search query q_i. (2) The search engine retrieves a set of relevant documents $\mathcal{D}_{k,i}$ from $\mathcal{C}$ based on q_i. These documents are then appended to the reasoning context. (3) The model processes the updated context to decide whether the accumulated information is sufficient to resolve the original query q. If the model determines it can answer, it generates the final answer a and terminates the trajectory; otherwise, it proceeds to the next reasoning step t_{i+1}. This iterative process continues until the model produces a terminal response or reaches a predefined maximum step limit N.

To optimize the LLM for multi-turn interaction with search engine, we treat the reasoning process as a sequential decision-making task and optimize it via reinforcement learning. We define the RL objective function for the policy LLM π_θ interacting with an external search engine $\mathcal{R}$ as follows:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{C}, y\sim\pi_\theta(\cdot|x;\mathcal{R})} \left[r_\phi(x, y)\right] - \beta\mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x; \mathcal{R}) \,||\, \pi_{\text{ref}}(y \mid x; \mathcal{R})\right], \quad (1)$$

where π_{ref} represents the frozen reference LLM, while r_ϕ and $\mathbb{D}_{\text{KL}}$ denote the reward function and the Kullback-Leibler divergence term, respectively. The variable x refers to input queries sampled from the corpus $\mathcal{C}$. The term y signifies the comprehensive reasoning trajectory, which consists of generated outputs interleaved with real-time results retrieved from the search engine $\mathcal{R}$. During training, these trajectories are sampled from the current policy $\pi_\theta(y|x; \mathcal{R})$, allowing the LLM to learn the optimal sequence of internal reasoning and external knowledge incorporation.

4 Method

In this section, we present Eval-RAR, an evaluation-driven reinforcement learning framework for retrieval-augmented reasoning. First, we extend the task definition introduced in the previous section into our proposed "Search-then-Evaluate" paradigm (Section 4.1), which formalizes how the LLM should evaluate—based on the reasoning of previous steps, the current query, and the newly retrieved documents—whether the original question

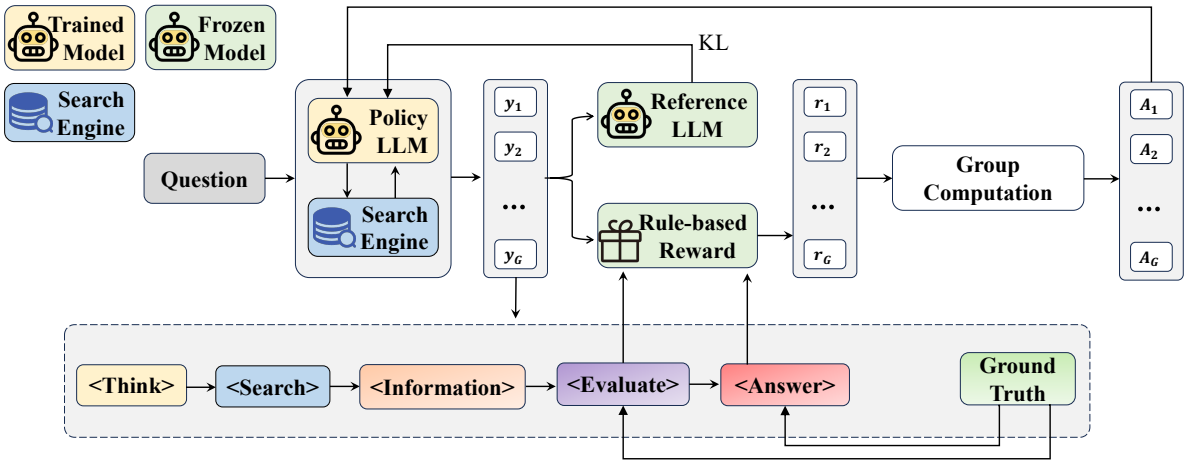


Figure 2: Overview of Eval-RAR. The figure illustrates the rollout of retrieval-enhanced GRPO with a search engine, which is guided by the evaluation reward and the outcome reward.

can be answered. Second, we detail the design of reward functions, including outcome rewards and evaluation rewards, followed by the training based on the GRPO RL algorithm (Section 4.2).

4.1 The Search-then-Evaluate Paradigm

Algorithm 1 Eval-RAR Workflow

Require: Input query x, policy model π_θ, retriever $\mathcal{R}$, maximum step limit N.
Ensure: Final response y.

Initialize response $y \leftarrow \emptyset$
Initialize step count $n \leftarrow 0$
while $n < N$ **do**
  Initialize current action rollout $y_n \leftarrow \emptyset$
  while True **do**
    Generate token: $y_t \sim \pi_\theta(\cdot \mid x, y + y_n)$
    Append y_t: $y_n \leftarrow y_n + y_t$
    if $y_t \in$ {</search>, </answer>, <eos>} **then**
      break
    end if
  end while
  $y \leftarrow y + y_n$
  if <search>...</search> detected in y_n **then**
    Query: $q_n \leftarrow$ Parse(<search>...</search>, y_n)
    Documents: $d_n \leftarrow \mathcal{R}(q_n)$
    $y \leftarrow y +$ <information>d_n</information>
    Evaluation: $y_{e_n} \sim \pi_\theta(\cdot \mid x, y)$ ▷ Evaluation step
    $y \leftarrow y + y_{e_n}$ ▷ y_{e_n} is <evaluate> e_n </evaluate>
  else if <answer>...</answer> detected in y_n **then**
    return Final response y
  else
    $y \leftarrow y +$"My action is wrong. Let me try again."
  end if
  $n \leftarrow n + 1$
end while
return Final response y

Building upon the task definition in Section 3 and inspired by Search-R1 (Jin et al., 2025), we represent each reasoning step t_i using a sequence of specialized functional tokens. In existing frameworks, a typical step is structured as:<think>... </think><search>...</search><information> ...</information>. If the model determines that the original query is resolvable, the final response is encapsulated within <answer>...</answer> special tokens. To strengthen this process, our proposed paradigm extends each step t_i by introducing an explicit evaluation phase: <think>... </think><search>...</search><information> ...</information><evaluate>...</evaluate> Within the <evaluate> block, the model is required to perform a self-evaluation of the current reasoning state derived from its previous thoughts and the newly retrieved information. The goal of this self-evaluation is to identify relevant supporting evidence or to recognize any critical information that is still missing. This deliberate evaluation step helps maintain coherence and factual soundness before the model proceeds to further search or finalize an answer. The complete workflow of this iterative multi-turn interaction is detailed in Algorithm 1.

To implement this structured reasoning process, we utilize a specialized system prompt (detailed in Appendix A.1) that instructs the model to adhere strictly to the interleaved execution of these special tokens.

4.2 Reward Design and Training

To optimize the model via RL, we employ a rule-based reward system that serves as the training signal. Our reward system is decomposed into two principal components: outcome reward and evalu-

ation reward. Specifically, outcome reward is designed to ensure the factual correctness of the final response. Evaluation reward is introduced to encourage the model to accurately identify and leverage relevant information during the self-evaluation phase, thereby supporting the overall reasoning integrity. These signals collectively guide the policy toward generating high-quality, grounded reasoning trajectories.

4.2.1 Outcome Reward

To ensure the ultimate accuracy of the model's reasoning process, we employ an outcome reward R_{ans}, which provides a sparse signal based on the correctness of the final output. The reward is then computed using an Exact Match (EM) criterion, represented by the indicator function $\mathbb{I}(\cdot)$ as:

$$R_{\text{ans}} = \mathbb{I}(\exists a \in \mathcal{A} : a = y_{ans}), \tag{2}$$

where y_{ans} is the extracted final answer from response y and $\mathcal{A}$ represents the set of ground-truth answers.

4.2.2 Evaluation Reward

We introduce an evaluation reward r_{eval} to explicitly encourage the model to identify and extract relevant evidence from its previous response. This reward is computed based on the content generated within the `<evaluate>` blocks. Specifically, we collect all self-evaluation steps across the reasoning trajectory and concatenate them into a single text sequence y_{eval}:

$$y_{eval} = \texttt{Concat}\left(\{e_n \mid e_n \in \mathcal{S}_{eval}(y)\}\right) \tag{3}$$

where $\mathcal{S}_{eval}(y)$ is the ordered set of all evaluation segments extracted from the reasoning trajectory y and `Concat` denotes the concatenation operator. The evaluation reward is then defined as:

$$R_{\text{eval}} = \begin{cases} r_{eval}, & \text{if } \mathbb{I}(\exists a \in \mathcal{A} : a \subseteq y_{eval}) > 0, \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where r_{eval} is a constant reward value.

4.2.3 Overall Reward

The total reward R_{total} is the combination of the individual components, providing a multi-level feedback signal to guide the reinforcement learning process. Formally, it is defined as follows:

$$R_{\text{total}} = \begin{cases} R_{\text{ans}}, & \text{if } R_{\text{ans}} > 0 \\ R_{\text{eval}}, & \text{otherwise} \end{cases} \tag{5}$$

This formulation prioritizes the correctness of the final answer while incentivizing the model to maintain structural discipline and identify critical evidence throughout the reasoning trajectory, even during failed attempts.

4.2.4 Training with GRPO

To optimize the unified generation policy π_θ over the interleaved reasoning and retrieval trajectories, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO eliminates the need for an additional value function approximator by using the relative rewards of multiple outputs sampled from the same input as a baseline. This significantly reduces computational overhead and stabilizes training in settings with sparse rewards.

For each input query x, we sample a group of G responses $\{y_1, y_2, \ldots, y_G\}$ from the current policy π_{old} with the retriever $\mathcal{R}$. The objective function for GRPO is defined as:

$$\begin{aligned} &J_{\text{GRPO}}(\theta) = \\ &\mathbb{E}_{x\sim\mathcal{C},\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|x;\mathcal{R})} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \\ &\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right) \\ &- \beta\mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\text{ref}}]\Bigg], \end{aligned} \tag{6}$$

where ϵ is the clipping ratio, β is the KL divergence penalty coefficient, π_{ref} is the reference policy and $\hat{A}_{i,t}$ represents the group-relative advantage. The $r_{i,t}(\theta)$ is formally defined as:

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t}; \mathcal{R})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t}; \mathcal{R})}. \tag{7}$$

We apply a retrieval token masking strategy. Since the documents are provided by the retriever and not generated by the model's policy, these tokens are masked during the loss computation.

5 Experiments

5.1 Experimental Setup

Datasets and metrics. We evaluate Eval-RAR across seven diverse QA benchmark datasets to assess its performance in scenarios with varying reasoning and retrieval complexity. These datasets are categorized into two groups: (1) **Single-Hop QA**, which focuses on factual inquiries typically requiring single-hop retrieval, including Natural

Methods	Single-Hop QA			Multi-Hop QA				
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Average
Reasoning without Retrieval								
Direct Inference	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
CoT	0.023	0.032	0.005	0.021	0.021	0.002	0.000	0.015
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-Base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
R1-Instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Reasoning with Retrieval								
Naive RAG	0.348	0.544	0.387	0.255	0.226	0.047	0.080	0.270
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	0.320	0.255
Reasoning with Retrieval via RL								
Search-R1-Base	0.421	0.583	0.413	0.297	0.274	0.066	0.128	0.312
Search-R1-Instruct	0.397	0.565	0.391	0.331	0.310	0.124	0.232	0.336
AutoRefine-Instruct	0.436	0.597	0.447	0.404	0.380	0.169	0.336	0.396
AutoRefine-Base	0.467	0.620	0.450	0.405	0.393	0.157	0.344	0.405
Eval-RAR-Instruct	0.462	0.611	0.448	0.403	0.383	**0.171**	0.352	0.404
Eval-RAR-Base	**0.471**	**0.629**	**0.453**	**0.412**	**0.394**	0.161	**0.368**	**0.413**

Table 1: Main results of Eval-RAR and baselines on QA benchmarks. The best performance is highlighted in **bold**.

Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); and (2) **Multi-Hop QA**, which requires multi-hop reasoning and evidence synthesis across multiple documents, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Following the setup of prior studies (Jin et al., 2025; Shi et al., 2025), we utilize a combined training set from NQ and HotpotQA to provide a balanced signal for both single-turn and multi-turn reasoning. We employ Exact Match (EM) as the evaluation metric.

Baselines. To comprehensively evaluate the effectiveness of our proposed method, we compare it against several representative baselines categorized by their reasoning and retrieval paradigms. **(1) Reasoning without Retrieval**: These methods rely on the model's parametric knowledge without retrieval. We include Direct Inference, where the model provides an answer without intermediate steps. Chain-of-Thought (CoT) (Wei et al., 2022), which prompts the model to generate a reasoning chain. SFT (Chung et al., 2024) using high-quality reasoning-and-retrieval trajectories and R1-based Training (DeepSeek-AI et al., 2025) where the model is trained via RL using only reasoning and answer steps without search engine access, providing a baseline for purely internal reasoning. **(2) Reasoning with Retrieval**: These approaches utilize external knowledge without additional training on search trajectories. Naive RAG (Lewis et al., 2020) performs a single-hop retrieval based on the initial query. IRCoT (Trivedi et al., 2023) interleaves CoT reasoning with retrieval to gather information iteratively. Search-o1 (Li et al., 2025b) represents an agentic search method that uses sophisticated prompting to manage multi-turn interactions. **(3) Reasoning with Retrieval via RL**: This category includes models optimized to interact with search engines via RL. We compare against Search-R1 (Jin et al., 2025), a recent state-of-the-art framework that trains the policy via GRPO. Building upon the Search-R1 paradigm, we also include AutoRefine (Shi et al., 2025) as a key baseline, which introduces a refinement stage to iteratively refine retrieved documents. All baseline results are taken from Search-R1 and AutoRefine.

Implementation details. Following the settings in Search-R1 (Jin et al., 2025), we conduct our experiments using both the Qwen2.5-3B-Base and Qwen2.5-3B-Instruct models (Qwen et al., 2025). To simulate a real-world search environment, we utilize the December 2018 Wikipedia

Methods	Single-Hop QA			Multi-Hop QA				
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Average
Eval-RAR-Instruct	0.462	0.611	0.448	0.403	0.383	**0.171**	0.352	0.404
w/o evaluation reward	0.423	0.583	0.429	0.341	0.336	0.111	0.248	0.353
Eval-RAR-Base	**0.471**	**0.629**	**0.453**	**0.412**	**0.394**	0.161	**0.368**	**0.413**
w/o evaluation reward	0.427	0.591	0.434	0.331	0.348	0.123	0.264	0.360

Table 2: Ablation results of Eval-RAR on QA benchmarks. The best performance is highlighted in **bold**.

dump (Karpukhin et al., 2020) as our external knowledge corpus and employ E5-base-v2 (Wang et al., 2022) as the search engine. The search engine is configured to retrieve the top-3 most relevant documents for each query. The r_{eval} is set to 0.1. The RL process is implemented using the verl framework (Sheng et al., 2025), and the model is trained for a total of 200 steps. For other training configurations, please refer to Appendix A.2.

5.2 Main results

Table 1 presents a performance comparison between the proposed Eval-RAR framework and a range of baseline methods across seven benchmarks. Analysis of the results yields three primary findings: (1) Eval-RAR consistently achieves state-of-the-art performance, outperforming both traditional Retrieval-Augmented Generation approaches and more recent Reinforcement Learning-based methods; (2) the framework demonstrates unique and significant advantages in complex, multi-step reasoning tasks; and (3) the learned reasoning policy exhibits high robustness across different model initializations and variants.

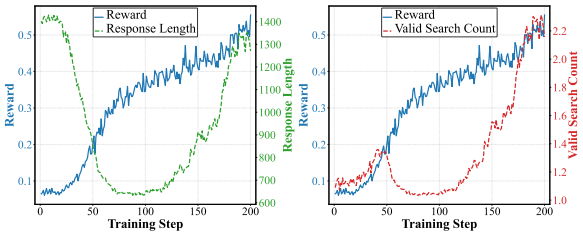
Superiority of the Eval-RAR. The Eval-RAR-Base variant achieves an overall average score of 0.413, marking a substantial improvement over the strong RL-based baseline Search-R1-Base and exceeding the performance of AutoRefine-Base. This consistent overall lead underscores the fundamental efficacy of the proposed "Search-then-Evaluate" paradigm. By explicitly equipping the model with a self-evaluation mechanism, the paradigm enables the systematic verification of information sufficiency and reasoning coherence before final answer generation, thereby mitigating the risks of premature commitment and insufficient grounding.

Pronounced Gains in Multi-hop Reasoning. While improvements are consistent across all categories, gains are significantly more pronounced in multi-hop QA compared to single-hop tasks. On complex benchmarks requiring evidence integration (e.g., Bamboogle, Musique), Eval-RAR exhibits a steeper improvement curve compared to Search-R1. This confirms that as reasoning chains lengthen, the ability to identify knowledge gaps becomes the primary bottleneck; by explicitly addressing these gaps, Eval-RAR prevents the logical drift that typically affects multi-turn RAG.

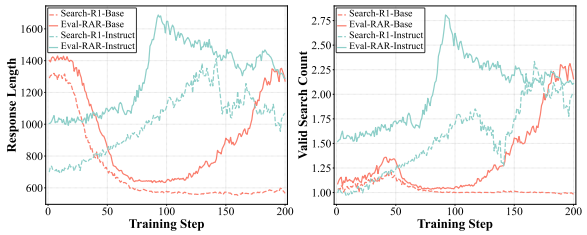
Robustness and Generalization Across Model Initializations. The effectiveness of Eval-RAR is consistently demonstrated across both Base and Instruct variants of the Qwen2.5-3B model. Notably, the Base model frequently achieves the highest scores, indicating that our GRPO-based reinforcement learning can autonomously cultivate self-evaluation and search guidance abilities without prior instruction tuning. This suggests that the performance improvements stem directly from our reward design and policy optimization rather than initial pre-trained capabilities.

5.3 Ablation Study

To investigate the contribution of our proposed fine-grained evaluation reward, we conduct an ablation study by comparing the full Eval-RAR framework against a variant that excludes this intermediate signal, denoted as *w/o evaluation reward*. In this setting, the model is trained solely using outcome-based rewards, relying on the final answer's correctness to optimize the entire reasoning-retrieval trajectory. As shown in Table 2, removing the evaluation reward leads to a significant performance decline across all benchmarks for both Base and Instruct models. Specifically, the average EM score drops from 0.413 to 0.360 for the Base model and from 0.404 to 0.353 for the Instruct model. The degradation is particularly acute in multi-hop datasets such as Bamboogle and Musique, where the reasoning chains are longer and more prone to error. This analysis confirms that while a self-evaluation phase is structurally present, the model



(a) Reward vs. Response Length and Valid Search Count



(b) Comparison of Search-R1 and Eval-RAR

Figure 3: Comparison based on response length and valid search count during RL training. (a) Trend comparisons of rewards versus response length and valid search count for the Eval-RAR-base model during training; (b) Trend comparisons of response length and valid search count during training for the base/instruct models of Search-R1 and Eval-RAR.

requires dense, intermediate supervision to effectively learn how to evaluate information utility. These results validate that the fine-grained evaluation reward is essential for mitigating reward sparsity and guiding the model toward a more rigorous and goal-aligned reasoning process.

6 Analysis

6.1 Reward vs. Response Length and Valid Search Count

To further understand the evolution of the Eval-RAR's behavior during the GRPO training process, we visualize the trends of reward, response length, and valid search count over the 200 training steps for the Qwen2.5-3B-Base model. As shown in Figure 3(a), reward exhibits a clear correlation with response length. During the initial stage (steps 0–75), response length drops sharply from about 1400 tokens to a minimum of 650 tokens, while the reward steadily rises—indicating that the model first learns to prune redundant or irrelevant reasoning steps. Beyond step 125, response length begins to rebound and eventually stabilizes. This U-shaped pattern suggests the model progressively learns to generate longer and more complete reasoning chains, which directly drives further reward growth. Parallel to this, reward also relates to the valid search count. Similar to the response length trend, the number of valid searches initially declines as the model optimizes its basic reasoning. After step 125, however, the count rises from a baseline of 1.1 to over 2.2 per trajectory by training end. This trend confirms that Eval-RAR effectively encourages the model to actively invoke the search engine to address knowledge gaps, thereby strengthening multi-hop retrieval performance.

6.2 Comparison of Search-R1 and Eval-RAR

To further validate the efficiency of our approach, we compare the training dynamics of Eval-RAR against the Search-R1 baseline. As illustrated in Figure 3(b), Eval-RAR consistently demonstrates superior learning stability and reasoning depth across both Base and Instruct configurations.

Reasoning Depth and Stability. As shown in the response length trajectory, both Eval-RAR variants successfully transition from an initial pruning phase to a significant increase in reasoning depth after step 125. Notably, Eval-RAR-Base exhibits a much smoother and more stable growth pattern than the Instruct variant, which shows higher variance. This suggests that our evaluation reward provides a more consistent signal for Base models to develop complex reasoning chains compared to instruction-tuned models.

Mastery of Multi-hop Retrieval. The advantage of our method is most evident in the Valid Search Count. While Search-R1-Base remains near a count of 1.0, failing to break out of single-turn patterns, Eval-RAR-Base displays a remarkably steady and progressive increase, eventually exceeding 2.25 searches per trajectory. The more stable progression of the Base model further indicates that the fine-grained evaluation reward is particularly effective at guiding raw models to develop structured, multi-step search strategies from scratch.

7 Conclusion

This paper presents Eval-RAR, a reinforcement learning framework that enhances retrieval-augmented reasoning via a "Search-then-Evaluate" paradigm. Through explicit self-evaluation, the model learns to assess information sufficiency and

identify knowledge gaps. We also propose a fine-grained evaluation reward to stabilize training and address the sparsity of outcome-based rewards. Experiments show that Eval-RAR outperforms state-of-the-art baselines, and further analysis reveals that it effectively masters multi-hop retrieval, demonstrated by increased search frequency and more thorough reasoning chains.

Limitations

Despite the notable performance improvements demonstrated by Eval-RAR, our work acknowledges two primary limitations. First, due to computational constraints, all experiments were conducted with 3B models. While the framework is conceptually scalable, its empirical effectiveness on significantly larger-scale models remains to be validated. Second, the explicit self-evaluation step introduces additional inference overhead. Future work will focus on optimizing the efficiency of the reasoning process to better balance performance and computational cost.

Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China (U25B2072), and the Fundamental Research Funds for the Central Universities (WK2150110032).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y. Wu, Yukun Li, Huazuo Gao, Shirong Ma, Wangding Zeng, Xiao Bi, Zihui Gu, Hanwei Xu, Damai Dai, Kai Dong, Liyue Zhang, Yishi Piao, and 21 others. 2024. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *Preprint*, arXiv:2406.11931.

Weibo Gao, Qi Liu, Linan Yue, Fangzhou Yao, Rui Lv, Zheng Zhang, Hao Wang, and Zhenya Huang. 2025. Agent4edu: Generating learner response data by generative agents for intelligent education systems. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 23923–23932. AAAI Press.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

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.

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

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rui Li, Liyang He, Qi Liu, Zheng Zhang, Heng Yu, Yuyang Ye, Linbo Zhu, and Yu Su. 2025a. UniRAG: Unified query understanding method for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14163–14178, Vienna, Austria. Association for Computational Linguistics.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. RA-DIT: retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2021. EKT: exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.*, 33(1):100–115.

Jinchang Luo, Mingquan Cheng, Fan Wan, Ni Li, Xiaoling Xia, Shuangshuang Tian, Tingcheng Bian, Haiwei Wang, Haohuan Fu, and Yan Tao. 2025. Globalrag: Enhancing global reasoning in multi-hop question answering via reinforcement learning. *Preprint*, arXiv:2510.20548.

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.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin

Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

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.

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.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Wenda Wei, Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Lixin Su, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Xueqi Cheng. 2025. Thinking forward and backward: Multi-objective reinforcement learning for retrieval-augmented reasoning. *Preprint*, arXiv:2511.09109.

Shicheng Xu, Liang Pang, Mo Yu, Fandong Meng, Huawei Shen, Xueqi Cheng, and Jie Zhou. 2024. Unsupervised information refinement training of large language models for retrieval-augmented generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 133–145, Bangkok, Thailand. Association for Computational Linguistics.

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

Heng Yu, Junfeng Kang, Rui Li, Qi Liu, Liyang He, Zhenya Huang, Shuanghong Shen, and Junyu Lu. 2025. CA-GAR: Context-aware alignment of LLM generation for document retrieval. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5836–5849, Vienna, Austria. Association for Computational Linguistics.

Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and Junbo Zhao. 2025a. LeTS: Learning to think-and-search via process-and-outcome reward hybridization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5109–5122, Suzhou, China. Association for Computational Linguistics.

Zheng Zhang, Ning Li, Qi Liu, Rui Li, Weibo Gao, Qingyang Mao, Zhenya Huang, Baosheng Yu, and Dacheng Tao. 2025b. The other side of the coin: Exploring fairness in retrieval-augmented generation. *Preprint*, arXiv:2504.12323.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. 2025a. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21816–21841, Suzhou, China. Association for Computational Linguistics.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025b. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *Preprint*, arXiv:2504.03160.

Yan Zhuang, Qi Liu, Zhenya Huang, Zhi Li, Binbin Jin, Haoyang Bi, Enhong Chen, and Shijin Wang. 2022. A robust computerized adaptive testing approach in educational question retrieval. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 416–426. ACM.

A Appendix

A.1 Instruction for Model Input

Prompt Template for Trajectory Generation

You are a helpful assistant excel at answering questions with multi-turn search engine calling. To answer questions, you must first reason through the available information using <think> and </think>. If you identify missing knowledge, you may issue a search request using <search> query </search> at any time. The retrieval system will provide you with the three most relevant documents enclosed in <information> and </information>. **After each search, you must conduct an evaluation, reviewing the question and the retrieved documents comprehensively to determine if the question can be answered. If the question can be answered, clearly cite the source content from the search documents that supports your answer. If the question cannot be answered, specify what information is missing: entity information, relational information, or specific time/location details. Provide your evaluation using <evaluate> and </evaluate>.** You may send multiple search requests if needed. Once you have sufficient information, provide a concise final answer using <answer> and </answer>. For example, <answer> Donald Trump </answer>. Question: `{question}`

A.2 Training Configuration

The training is performed on a cluster of four NVIDIA A100-80GB GPUs. We use a total training batch size of 512, a validation batch size of 256, and a micro-batch size of 16, with the policy model's learning rate set to 1×10^{-6}. For the GRPO group sampling, we generate $G = 5$ responses per prompt. The KL divergence regularization coefficient is set to $\beta = 0.001$, and the clipping ratio is $\epsilon = 0.2$. To maximize memory efficiency, we enable gradient checkpointing and utilize Fully Sharded Data Parallel (FSDP) with CPU offloading for parameters, gradients, and optimizer states. For high-throughput rollout sampling, we integrate vLLM (Kwon et al., 2023) with a tensor parallel size of 1 and a GPU memory utilization ratio of 0.6.

A.3 Case Study

Question: When did the person who gave the Checkers speech die?
Ground Truth: April 22, 1994
Search-R1: <think>The Checkers speech was given by Richard Nixon in 1952 as a campaign event. Nixon was the Republican candidate for president at the time.</think> <search>person who gave the Checkers speech died</search> <information>Doc 1(Title: "Checkers speech") know why,"" which contributed to the failure of his 1960 presidential run ... Doc 2(Title: "Checkers speech") Checkers speech **The Checkers speech or Fund speech was an address made on September 23, 1952** ... Doc 3(Title: "Checkers speech") in later years, Hal Bochin (who wrote a book about Nixon's rhetoric) suggests that Nixon succeeded at the time ...</information> <answer>**September 23, 1952**</answer>
Eval-RAR: <think>The target question refers to a historical figure who delivered a speech involving a 'Checkers' reference. I need to identify this person first and then find out when they passed away.</think> <search>When did the person who gave the Checkers speech die</search> <information>Doc 1(Title: "Checkers speech") know why,"" which contributed to the failure of his 1960 presidential run ... Doc 2(Title: "Checkers speech") Checkers speech **The Checkers speech or Fund speech was an address made on September 23, 1952, by the Republican candidate for Vice President of the United States, California Senator Richard Nixon.** ... Doc 3(Title: "Checkers speech") in later years, Hal Bochin (who wrote a book about Nixon's rhetoric) suggests that Nixon succeeded at the time ...</information> <evaluate>From the information gathered, **the target person delivering the Checkers speech is Richard Nixon. Now, the main concern is to find the death date of Richard Nixon.**</evaluate> <search>When did Richard Nixon die</search> <information>Doc 1(Title: "Death and state funeral of Richard Nixon") **Death and state funeral of Richard Nixon On April 22, 1994** ... Doc 2(Title: "Death and state funeral of Richard Nixon") brain ... Doc 3(Title: "Richard Nixon") Hospital–Cornell Medical Center in Manhattan ...</information> <evaluate> **With the target person identified as Richard Nixon and the date of death confirmed, we can now answer the target question.**</evaluate> <answer>**April 22, 1994**</answer>

Table 3: Case study: **Green** for correct answer, **Red** for incorrect.

A.4 Dataset Statistics

Table 4 lists the dataset statistics used in this work. The training set for Eval-RAR combines NQ and HotpotQA's train splits, comprising 169,615 examples. The evaluation set, which includes 51,713 examples, is composed of the test splits from NQ, TriviaQA, PopQA, and Bamboogle, supplemented by the development splits from HotpotQA, 2Wiki,

and Musique where test data was not provided.

Dataset	Train	Dev	Test
Single-hop QA			
NQ	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313
PopQA	-	-	14,267
Multi-hop QA			
HotpotQA	90,447	7,405	-
2Wiki	15,000	12,576	-
Musique	19,938	2,417	-
Bamboogle	-	-	125

Table 4: Statistics of the seven datasets.