

# R-Search: Empowering LLM Reasoning with Search via Multi-Reward Reinforcement Learning

Qingfei Zhao<sup>1,2</sup>, Ruobing Wang<sup>1,2</sup>, Dingling Xu<sup>3</sup>, Daren Zha<sup>1\*</sup>, Bowen Ma<sup>4</sup>, Zhichun Wang<sup>3</sup>, Shijie Jia<sup>1</sup>, Limin Liu<sup>1</sup>, Xin Wang<sup>1</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences;  
<sup>2</sup>School of Cyber Security, University of Chinese Academy of Sciences;  
<sup>3</sup>Beijing Normal University; <sup>4</sup>Minzu University of China  
{zhaoqingfei, wangruobing, zhadaren}@iie.ac.cn

## Abstract

Large language models (LLMs) have notably progressed in multi-step and long-chain reasoning. However, extending their reasoning capabilities to encompass deep interactions with search remains a non-trivial challenge, as models often fail to identify optimal reasoning-search interaction trajectories, resulting in sub-optimal responses. We propose **R-Search**, a novel reinforcement learning framework for Reasoning-Search integration, designed to enable LLMs to autonomously execute multi-step reasoning with deep search interaction, and learn optimal reasoning-search interaction trajectories via multi-reward signals, improving response quality in complex logic- and knowledge-intensive tasks. R-Search guides the LLM to dynamically decide when to search or reason, while globally integrating key evidence to enhance deep knowledge interaction between reasoning and search. During RL training, R-Search provides multi-type rewards to jointly optimize the reasoning-search trajectory. Experiments on seven datasets show that R-Search significantly outperforms mainstream RAG baselines. The code and data are available at <https://github.com/QingFei1/R-Search>.

## 1 Introduction

Large language models (LLMs) have demonstrated substantial progress across a wide range of natural language processing (NLP) tasks, driven by their impressive language understanding and reasoning abilities (OpenAI, 2023; Plaat et al., 2026). In logic-intensive tasks (Asai and Hajishirzi, 2020; Shao et al., 2024), state-of-the-art LLMs, exemplified by DeepSeek-R1 (DeepSeek-AI, 2025), have demonstrated remarkable capabilities in long-chain and multi-step reasoning (Jaech et al., 2024). In knowledge-intensive tasks (Lewis et al., 2020;

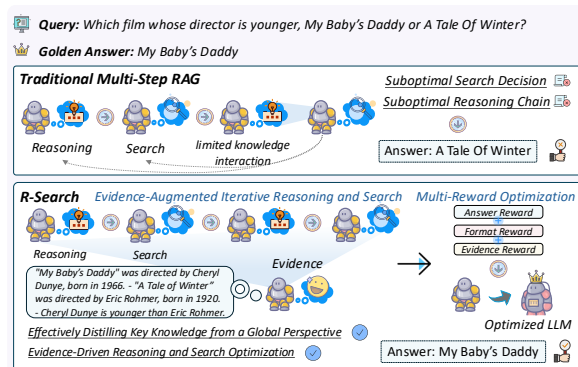


Figure 1: A comparison between traditional multi-step RAG and R-Search.

Trivedi et al., 2023), even LLMs with strong reasoning capabilities are susceptible to generating hallucinated outputs (Zhang et al., 2023). This primarily arises from inherent limitations in the accuracy, timeliness, and coverage of their parametric knowledge. To mitigate hallucination, LLM-based Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Zhao et al., 2024; Kim et al., 2024) incorporates search actions before generation, enabling the LLM to augment its input with non-parametric knowledge in textual form. This allows the LLM to flexibly access and integrate relevant information from external knowledge sources, thereby enhancing the reliability of downstream generation. However, downstream generation often struggles to benefit from reasoning alone or one-time search when addressing more complex logic- and knowledge-intensive tasks, e.g., multi-hop question-answering (QA) task (Yang et al., 2018). In tackling such complex tasks, the LLM is expected to dynamically integrate external knowledge into the reasoning process, not only to bridge the knowledge gap but to guide and deepen the reasoning trajectory.

Previous multi-step RAG methods (Jeong et al., 2024; Trivedi et al., 2023) enable the integration of

\*Corresponding author

external knowledge into reasoning by prompting LLMs to iteratively perform reasoning-search interactions. As depicted in Figure 1, in this process, retrieved information enhances the model’s reasoning, which in turn guides the subsequent search, forming a dynamic loop between them. However, these methods typically rely on the LLM’s internal cognition to decide when and what to retrieve, leading to two main limitations: **1)** The retrieval timing determined by the model’s internal knowledge distribution does not always align with the actual need for retrieval; **2)** the modular and decoupled design of reasoning and search limits deep interaction of external knowledge into the reasoning chain. As a result, the model often makes decisions based only on partial information from previous searches or thoughts. These limitations lead to suboptimal or even incorrect reasoning-search trajectories, ultimately reducing the quality of the final outputs.

To this end, we propose R-Search, a novel reinforcement learning (RL)-based framework that enables LLMs to dynamically interleave multi-step reasoning and search, and to learn optimal reasoning-search trajectories through multi-reward signals. **First**, R-Search allows the LLM to trigger retrieval dynamically at the token level, seamlessly integrating retrieved content into the reasoning process for deeper coupling between reasoning and external knowledge. After the interaction, the LLM distills retrieved documents into evidence through reasoning. This facilitates the LLM in re-evaluating and structuring critical knowledge from a global perspective, thereby enhancing its focus on the facts most pertinent to solving the task. **Second**, we design a multi-type reward mechanism that incorporates answer quality, evidence quality, and format correctness as reward signals. These complementary signals promote the model to learn the optimal reasoning-search interaction sequence. In particular, the evidence reward encourages the model to focus on the factual quality of key intermediate reasoning steps, promoting more robust reasoning paths and reducing the risk of shortcut-driven or speculative behavior.

To summarize, our contributions are threefold:

**(1) Framework Design:** We propose R-Search, a novel RL-based framework that jointly optimizes complex reasoning-search trajectories in RAG. R-Search promotes robust policy learning by interleaving evidence-augmented multi-step reasoning with dynamic search and optimizing through multi-reward modeling. It effectively guides the LLM to

ensure both the soundness of intermediate reasoning and the completeness of retrieved knowledge.

**(2) Superior Performance and Insightful Analysis:** We conduct extensive experiments on seven datasets across both multi-hop and single-hop QA tasks, demonstrating the superiority of R-Search over Vanilla and advanced RAG baselines. Further analyses through ablation studies and training dynamics validate the effectiveness of evidence integration and multi-reward optimization, and provide insights into characteristics of knowledge in evidence and search behavior.

**(3) R-Search-as-a-Tool (R-Search-Tool):** We propose R-Search-Tool, which enriches downstream generation models with distilled, reasoning-driven knowledge snippets. R-Search-Tool serves as a transferable cognitive scaffold that enhances answer quality and knowledge generalization.

## 2 Related Work

### 2.1 Retrieval-Augmented Generation

LLM-based RAG systems (Fan et al., 2024; Ram et al., 2023; Yoran et al., 2024) have demonstrated significant performance gains across various NLP tasks (Lewis et al., 2020; Guan et al., 2025; Brown et al., 2020; OpenAI, 2023; Plaat et al., 2026), especially in open-domain question answering. As tasks involving more complex reasoning, e.g., multi-hop QA tasks (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023), traditional RAG methods (Lewis et al., 2020; Guu et al., 2020; Sachan et al., 2021) struggle with insufficient external knowledge integration. Several advanced RAG methods (Jeong et al., 2024; Jiang et al., 2023; Cheng et al., 2024; Trivedi et al., 2023; Wang et al., 2025; Shao et al., 2023; Asai et al., 2024) attempt to enable deeper knowledge exploration through adaptive or multi-step reasoning-search interactions. For instance, Search-o1 (Li et al., 2025) integrates deep search into its reasoning process through prompt engineering.

Nevertheless, these methods still heavily rely on carefully crafted prompts, making them difficult to scale and limiting the depth of interaction between search and reasoning, resulting in suboptimal answers. In our work, we aim to construct an RL-based RAG pipeline that enables flexible reasoning-search interaction and optimizes interaction trajectories.

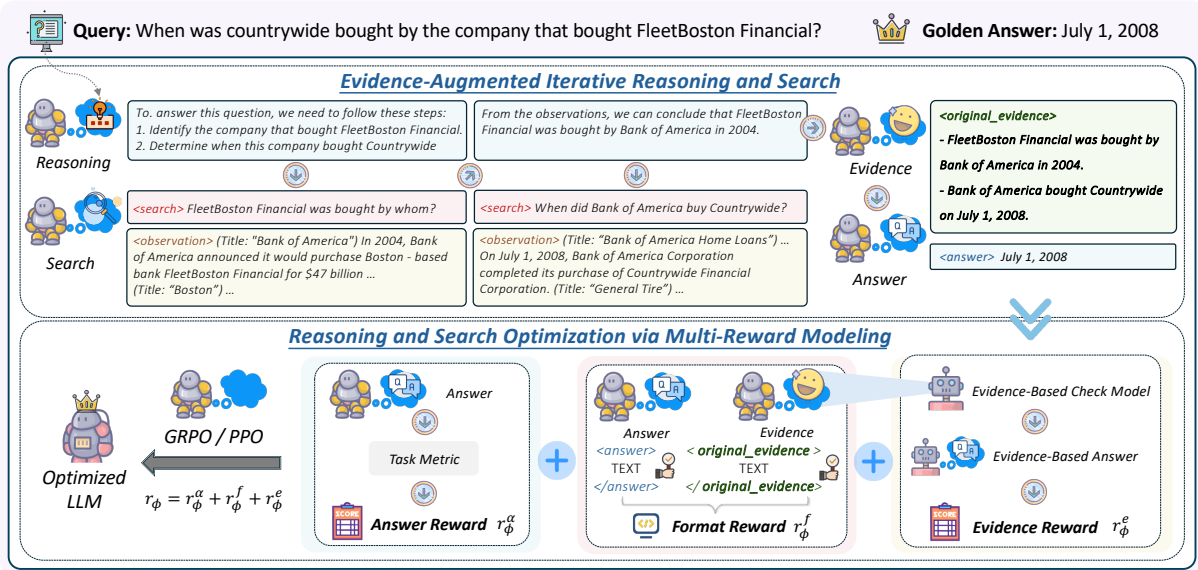


Figure 2: An overview of the R-Search framework.

## 2.2 Reinforcement Learning for RAG

Reinforcement learning (RL) (Kaelbling et al., 1996; Wiering and Van Otterlo, 2012) is an effective paradigm for enhancing the reasoning capabilities of LLMs. Recent studies (DeepSeek-AI, 2025; Shao et al., 2024; Jaech et al., 2024) have shown that RL with rule-based reward functions enables models to acquire complex task reasoning and self-correction abilities (Weng et al., 2023; Kumar et al., 2025) without explicit intermediate supervision. In particular, RL algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) have demonstrated the potential to improve LLMs’ autonomous search and reasoning abilities.

However, most existing RL approaches primarily focus on internal reasoning, with limited integration of external search information and insufficient handling of multi-turn interactions between reasoning and search. Moreover, the design of reward functions specifically tailored for RAG scenarios has not been sufficiently explored. Search-R1 (Jin et al., 2025a) and R1-Searcher (Song et al., 2025) rely on answer correctness as a reward, which often results in sparse feedback (Zhang and Zuo, 2025) and insufficient attention to the reasoning process, thereby increasing the training difficulty. To address these challenges, our work further explores multi-reward RL-based RAG methods that enable LLMs to perform more autonomous exploration, achieve more robust reasoning, and establish

deeper interactions between search and reasoning.

## 3 R-Search

In this section, we present the R-Search RAG pipeline in its formalized decision-process form. We then elaborate on the evidence-augmented reasoning and the optimization procedure via multi-reward modeling, as illustrated in Figure 2.

### 3.1 Preliminary of R-Search

R-Search tightly interweaves reasoning and search through several LLM-based components, enabling an adaptive and active process for exploring external knowledge to solve complex questions. In this process, the LLM can determine its next action based on its current observation of the environment, such as the reasoning trajectory and the retrieved external knowledge observed so far. The possible actions include initiating a new search, continuing the reasoning process, or generating the final answer. We formalize the RAG pipeline of R-Search as follows.

$$a_1 \rightsquigarrow a_2 \rightsquigarrow \dots \rightsquigarrow a_T, \quad a_t \in \{\mathcal{S}, \mathcal{R}\}, \quad (1)$$

where  $\mathcal{S}$  and  $\mathcal{R}$  denote the search and reasoning actions  $a$ , respectively, and  $T$  is the total number of decision time steps in the action sequence. Such a complex RAG pipeline can be viewed as a Partially Observable Markov Decision Process. Therefore, we develop a novel RL-based framework that optimizes the multi-step interleaved trajectories of reasoning and search.

### 3.2 Evidence-Augmented Iterative Reasoning and Search

Figure 5 describes the rollout template (more details in Appendix B). In the rollout process, we first prompt the LLM to generate a long Chain-of-Thought (CoT) based on the original query  $q$ , thereby constructing an explicit reasoning process. During the reasoning process, we encourage the model to trigger the search action at appropriate points to acquire external non-parametric information. Whenever the model determines that search is needed at the current reasoning state, it generates a new search query enclosed within `<search>` and `</search>` tags. We identify this specific search token and feed the generated query  $q^*$  into a search tool to retrieve top- $k$  relevant documents  $D_k = \{d_1, d_2, \dots, d_k\}$ . These documents are wrapped with specific tokens `<observation>` and `</observation>`, and appended to the existing reasoning trajectory. The LLM then re-engages the reasoning process, ultimately forming an interactive loop between reasoning and search.

Next, when the LLM determines that the current state is sufficiently informative to produce a final answer  $\alpha$ , we prompt it to rethink all previously retrieved information and derive factual evidence  $e$  supporting the resolution of the original query  $q$ . This enables the LLM to reason from a global perspective, leveraging all potentially relevant knowledge observed from the external information environment to support answer generation. Moreover, we leverage the model’s internal reasoning capabilities to interpret and integrate external knowledge during the generation of factual evidence. We wrap the factual evidence with the special tokens `<original_evidence>` and `</original_evidence>`. After the evidence generation, the LLM continues to output the final answer  $\alpha$  within the specific tokens box `<answer>` and `</answer>`.

### 3.3 Reasoning and Search Optimization via Multi-Reward Modeling

Due to the high cost and potential bias associated with training reward models using human feedback, we follow (Mu et al., 2024) and adopt the rule-based reward. Considering the complementary effects of multiple rewards (Dann et al., 2023), we design multi-type reward signals  $r_\phi$ , including evidence reward  $r_\phi^e$ , answer reward  $r_\phi^\alpha$ , and format reward  $r_\phi^f$ .

**Answer Reward.** Metrics such as F1-Score (F1) and Exact Match (EM) are widely used to evaluate the correctness of model outputs. We choose the moderately strict F1-score to construct the answer reward signal. Specifically, we extract the content  $\alpha_{\text{pred}}$  within `<answer>` and `</answer>` tags generated by the policy model  $\pi_\theta$  and compute the F1-score against the gold answer  $\alpha_{\text{gold}}$ , which serves as the answer reward  $r_\phi^\alpha$ .

$$r_\phi^\alpha(q, o) = \frac{2 \cdot |\alpha_{\text{pred}} \cap \alpha_{\text{gold}}|}{|\alpha_{\text{pred}}| + |\alpha_{\text{gold}}|} \quad (2)$$

where  $|\alpha_{\text{pred}} \cap \alpha_{\text{gold}}|$  is the number of word-level overlaps between the predicted and gold answers.

**Evidence Reward.** Due to the quality of evidence directly impacting answer accuracy, we introduce an evidence-based check model  $\pi_c$  to check the quality of knowledge in evidence. We then use the frozen  $\pi_c$  to construct an evidence reward computation pipeline that operates on the shared evidence but performs independent reasoning (evidence template in Appendix B).

$$\alpha_c \sim \pi_c(\cdot | q, e), r_\phi^e(q, o) = \text{F1}(\alpha_c, \alpha_{\text{gold}}) \quad (3)$$

First, the  $\pi_c$  shares the same evidence  $e$  with  $\pi_\theta$  and generates an evidence-based answer  $\alpha_c$  based on  $q$ . We then apply the same answer reward computation process to  $\alpha_c$  to compute  $r_\phi^e$ . The usage of a separate check model, rather than reusing the policy model to evaluate evidence quality, facilitates the mitigation of answer bias introduced by the policy model’s inherent preferences. As a result, the evidence reward signal more objectively reflects the underlying factual quality of the evidence.

**Format Reward.** Format reward ensures that generated content adheres to structural conventions and remains parseable for downstream use (DeepSeek-AI, 2025). Specifically, we enforce that the evidence appears in exactly one box, marked by `<original_evidence>` and `</original_evidence>`. Similarly, we require the final answer  $\alpha$  to be enclosed in exactly one `<answer>` and `</answer>` box. We formalize the calculation pipeline of the format reward  $r_\phi^f$ :

$$r_\phi^f(q, o) = \gamma_\alpha^f \mathbb{I}_A + \gamma_e^f \mathbb{I}_S \mathbb{I}_E. \quad (4)$$

where  $\mathbb{I}_S$ ,  $\mathbb{I}_A$ , and  $\mathbb{I}_E$  are indicator functions denoting whether retrieval is triggered, the answer is well-formatted, and the evidence is well-formatted, respectively.  $\gamma_\alpha^f$  and  $\gamma_e^f$  are the reward values for

answer formatting and evidence formatting, respectively. We present the formula for the overall reward  $r_\phi$ :

$$\begin{aligned} r_\phi &= r_\phi^\alpha + r_\phi^e + r_\phi^f \\ \text{s.t. } r_\phi^\alpha, r_\phi^e &\in [0, 1], \\ r_\phi^f &\in \{0, \gamma_e^f, \gamma_\alpha^f, \gamma_e^f + \gamma_\alpha^f\} \end{aligned} \quad (5)$$

**Multi-Reward Optimization.** To guide the model in continuously improving its reasoning and search capabilities, we fine-tune the policy model using GRPO and PPO by maximizing the expected reward over multi-step inference and search trajectories. The optimization objective is defined as:

$$\begin{aligned} \mathcal{J}(\theta) &= \mathbb{E}_{q \sim \mathcal{D}, o \sim \pi_\theta(\cdot | q; \mathcal{S})} [r_\phi(q, o)] \\ &\quad - \beta \mathbb{D}_{\text{KL}}[\pi_\theta(o | q; \mathcal{S}) \parallel \pi_{\text{ref}}(o | q; \mathcal{S})] \end{aligned} \quad (6)$$

where  $\pi_\theta$  is the policy model,  $q$  and  $o$  are the input question and generated output, and  $\mathcal{S}$  denotes search mechanism. Here,  $r_\phi$  is the reward function,  $\pi_{\text{ref}}$  the frozen reference model, and  $\beta$  the KL penalty coefficient. Unlike  $\pi_\theta(\cdot | q)$ , the policy  $\pi_\theta(\cdot | q; \mathcal{S})$  executes an interleaved process of reasoning-search to generate the rollout sequence, denoted as **Reason**  $\bowtie$  **Search**.

The rollout sequence consists of both LLM-generated tokens and tokens originating from external search, i.e., non-generated observations. During loss computation, we include only the LLM-generated tokens, i.e., evidence, final answers, search queries, and other intermediate reasoning steps, while excluding non-parametric observations retrieved from external knowledge sources. This selective loss design ensures that evidence participates in training and provides useful learning signals for the model to improve its ability to select, interpret, and integrate external knowledge. Meanwhile, it avoids introducing gradient noise from non-generated tokens, which may misalign the training dynamics with the intended optimization objective (Jin et al., 2025a). Importantly, although the evidence is conditioned on retrieved content in multi-step reasoning and search interactions, it is ultimately sampled from the model’s policy distribution, making it eligible for the gradient update.

## 4 Experimental Setup

### 4.1 Datasets, Metrics and Search Tools

**Datasets.** We conduct extensive experiments on seven datasets, covering both complex multi-hop

and simpler single-hop QA tasks. The multi-hop QA serves to evaluate whether R-Search can handle complex logic- and knowledge-intensive questions. The single-hop QA assesses its ability to address knowledge-intensive questions and explore its robustness across questions with varying levels of complexity. For multi-hop task, we adopt four challenging datasets: **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA (2WikiMQA)** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022), and **Bamboogle** (Press et al., 2023). For single-hop task, we select three factoid-based QA datasets, including **NQ** (Kwiatkowski et al., 2019), **PopQA** (Mallen et al., 2023), and **TriviaQA** (Joshi et al., 2017). The dataset characteristics, versions, and sizes are provided in Appendix A.

**Evaluation Metrics.** Following existing work (Shao et al., 2023), we adopt two standard evaluation metrics for QA tasks: F1 and EM for all datasets. EM is a more stringent metric than F1, as it measures string-level exact matches between the normalized prediction and the golden answer.

**Search Tools.** Effective search actions require appropriate sources and approaches. We use a dense retriever with the E5 model for all datasets. For single-hop QA datasets and Bamboogle, we use the 2018 Wikipedia dump as the corpus for retrieving open-domain knowledge. For the remaining three multi-hop datasets, we use the Wikipedia corpora version released by (Trivedi et al., 2023), each aligned with its corresponding dataset.

### 4.2 Baselines and Backbone LLMs

Our experiments compare against eight baselines. In No Search (NS), we evaluate the ability of LLMs to answer questions using only their internal parametric knowledge. Vanilla RAG extends NS by adding a one-time search step, forming the simplest retrieval-and-generation pipeline. Furthermore, we compare with Multi-Step RAG (MSRAG) methods, where approaches such as Iter-Retgen(Shao et al., 2023) and IRCOT(Trivedi et al., 2023) focus on iterative reasoning and search, whereas others like Adaptive-RAG (Jeong et al., 2024) and Search-o1 achieve more flexible, agentic reasoning-search interactions. We also compare with Search-R1 and R1-Searcher, both of which are RL-based methods designed to improve the reasoning-search interaction capability of LLMs.

For backbone LLMs, we train two open-source models via RL, i.e., **Qwen-2.5-3B/7B** (Yang et al., 2024). To mitigate the influence of the model’s

Method	Multi-Hop QA								Single-Hop QA						Avg.
	2WikiMQA*		HotpotQA†		MuSiQue†		Bamboogle†		NQ†		TriviaQA†		PopQA†		
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	
<i>Qwen-2.5-3B</i>															
No Search	24.4	27.8	15.0	20.6	1.4	7.2	2.4	9.5	9.8	17.8	32.0	37.7	12.8	16.4	16.8
Vanilla RAG	33.6	38.0	34.0	43.2	5.6	11.7	9.6	19.6	37.2	46.7	58.0	66.8	39.4	46.7	35.0
Iter-Retgen	33.2	38.2	34.4	43.4	8.2	14.8	12.0	20.2	38.0	47.4	60.2	68.5	43.2	49.4	36.5
IRCoT	35.8	46.0	39.0	50.4	9.2	17.6	23.2	33.3	23.4	35.4	45.8	56.5	31.6	41.5	34.9
Adaptive-RAG	35.0	43.9	38.0	49.0	25.4	35.6	24.0	32.8	37.2	46.7	55.4	64.4	35.8	43.8	38.0
Search-o1	23.2	32.2	15.0	22.7	4.4	14.0	21.6	31.8	24.6	37.3	48.4	58.3	27.0	36.3	28.3
Search-R1	58.8	68.1	46.2	57.8	24.4	32.9	41.6	53.9	34.4	44.1	56.6	63.2	37.0	43.5	47.3
<b>R-Search</b>	65.0	72.6	43.4	54.4	25.8	34.8	37.6	49.8	35.2	46.0	56.0	64.0	37.0	44.9	47.6
<b>R-Search-Tool</b>	66.0	74.4	48.0	59.4	27.2	36.6	39.2	51.4	35.6	47.2	59.6	68.2	39.8	47.7	50.0
<i>Qwen-2.5-7B</i>															
No Search	23.8	28.1	19.6	26.7	3.8	11.3	11.2	19.7	13.8	21.9	46.0	52.2	15.6	19.6	22.4
Vanilla RAG	35.4	40.5	37.4	48.1	7.2	14.6	20.8	29.7	35.0	46.7	60.0	68.5	37.6	47.6	37.8
Iter-Retgen	37.4	43.4	42.8	53.1	10.6	19.9	22.4	31.0	37.8	48.4	61.2	69.5	38.6	46.8	40.2
IRCoT	34.2	45.5	40.4	53.7	9.0	17.3	20.0	32.3	19.6	35.5	55.2	66.2	33.0	43.6	36.1
Adaptive-RAG	33.8	42.4	42.4	55.3	9.0	16.9	20.8	32.5	35.0	46.7	58.8	67.4	35.8	45.7	38.8
Search-o1	33.0	42.1	20.6	30.1	11.0	20.5	35.2	44.8	21.0	34.1	55.4	66.1	27.6	35.4	34.1
R1-Searcher	63.6	73.9	<b>58.0</b>	<b>70.1</b>	29.6	41.1	45.6	55.1	38.2	<b>49.8</b>	60.4	69.1	36.0	43.2	52.4
Search-R1	67.0	75.4	48.4	60.9	25.8	36.2	<b>47.2</b>	<b>58.4</b>	<b>39.8</b>	49.1	<b>65.0</b>	70.8	<u>41.0</u>	46.9	52.3
<b>R-Search</b>	<b>69.8</b>	<u>77.7</u>	52.2	64.4	<b>31.4</b>	<u>41.6</u>	42.4	57.6	<u>38.0</u>	49.1	<u>64.2</u>	<u>71.7</u>	<b>41.8</b>	<b>48.1</b>	<u>53.6</u>
<b>R-Search-Tool</b>	<u>67.6</u>	<b>78.0</b>	<u>54.6</u>	<u>66.6</u>	<u>31.2</u>	<b>42.6</b>	44.8	<b>59.7</b>	37.6	<u>49.3</u>	63.2	<b>72.2</b>	40.4	<u>48.0</u>	<b>54.0</b>

Table 1: Overall performance results (%). **Bold** and underline represent the best and second-best results. \* and † are in-domain and out-of-domain datasets. R-Search-Tool employs GLM-4-Plus (Zeng et al., 2024) for final generation.

internal knowledge on evidence checking, we adopt small-scale models, including Llama-3.2-3B (Meta, 2024) and Qwen-2.5-3B, in the evidence reward calculation process. Unless otherwise specified, all models used in our experiments correspond to instruction-tuned variants.

### 4.3 Implementation Details

During evaluation, we align the top- $k$  retrieval setting to 5 across all methods to ensure fair comparison. We also employ vLLM (Kwon et al., 2023) to accelerate inference. For GRPO and PPO training, we use only the 2WikiMQA training set and train on 8xA100 80GB GPUs. Primary training hyperparameters include a maximum total of 195 training steps and a batch size of 256. We set the retrieval top- $k$  during training to 3, and configure the learning rate and warm-up ratio to  $1e-6$  and 0.95, respectively. During the rollout process, we sample 5 responses for each input prompt, with the KL divergence coefficient  $\beta$  set to 0.001, and fix both  $\gamma_e^f$  and  $\gamma_\alpha^f$  to 0.2.

## 5 Results and Analysis

### 5.1 Overall Performance

In Table 1, we evaluate R-Search on seven benchmark datasets against eight baselines, confirming its performance advantages. R-Search achieves stable performance improvements across parameter scales ranging from 3B to 7B. It attains the best performance on the in-domain 2WikiMQA dataset and demonstrates overall improvements on out-of-domain datasets, indicating strong generalization. Moreover, the R-Search-Tool achieves the best average performance, validating both the effectiveness of our training approach and the feasibility of offloading complex reasoning-search interactions from powerful black-box models to locally deployable small-scale models.

**R-Search facilitates deep knowledge exploration.** R-Search consistently outperforms Vanilla RAG and No Search. In particular, R-Search excels at handling questions with long reasoning chains and complex retrieval requirements, achieving an

improvement of up to 27.0 F1 points over Vanilla RAG on MuSiQue. These results indicate that R-Search effectively supports deep knowledge exploration and maintains a stable reasoning-search interaction, enabling robust performance across questions of varying complexity.

**R-Search effectively optimizes the interaction between search and reasoning through multi-reward signals, enabling stronger performance gains on highly complex tasks.** We compare our method with MSRAG methods to evaluate its performance advantages. Experimental results demonstrate that our method significantly outperforms these MSRAG methods, achieving an average performance improvement of up to 13.4 points. Importantly, we observe that although multi-step RAG methods alleviate the knowledge limitations of vanilla RAG by performing multiple searches, their overall performance remains unstable. This instability mainly arises from the frequent mismatch between the model’s internal knowledge reasoning and the actual retrieval requirements during the intertwined reasoning and search process, which may introduce irrelevant passages. Furthermore, we evaluate our method against RAG+RL baselines, such as Search-R1 and R1-Searcher. The results show that our method, R-Search, achieves substantially better overall performance than both Search-R1 and R1-Searcher. Notably, on the highly challenging MuSiQue dataset, R-Search outperforms Search-R1 by 5.4 F1 points. We attribute this improvement to the incorporation of evidence integration and multi-reward modeling in the RL process, which strengthens knowledge integration across steps and distills key evidential information within the reasoning chain (see case study in Appendix C). This design supports a more thorough and effective reasoning-search procedure, making R-Search particularly well-suited to tasks with complex reasoning and retrieval requirements.

**Beyond end-to-end answer generation, R-Search can also function as a transferable knowledge tool, termed R-Search-as-a-Tool (R-Search-Tool).** We further investigate the effectiveness and transferability of the knowledge contained in the evidence generated by R-Search. To this end, we evaluate evidence quality by injecting the same evidence into downstream generative models of varying scales and comparing their answer quality, thereby isolating the contribution of the evidence itself from generation model capacity. As illustrated in Figure 3, the evidence consistently

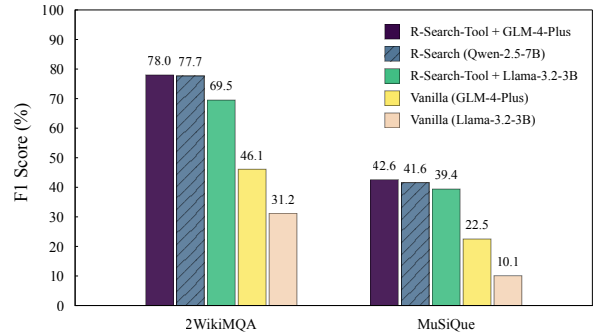


Figure 3: R-Search-Tool. RSTool+<model> refers to using the optimized model (Qwen-2.5-7B in this case) to produce evidence, which is subsequently fed as contextual input to other downstream generation models.

	Multi-Hop QA				Single-Hop QA		
	Hot.	2Wiki.	MuSi.	Bamb.	NQ	Tri.	Pop.
<i>Impact of Evidence</i>							
R-Search	<b>64.4</b>	<b>77.7</b>	<b>41.6</b>	<b>57.6</b>	<b>49.1</b>	<b>71.7</b>	<b>48.1</b>
w/o $R^e$	53.7	76.5	35.6	50.0	48.2	66.6	46.1
w/o Evidence	61.9	77.5	39.6	55.9	47.3	70.2	48.0
<i>Impact of Different Evidence-Based Check Models</i>							
Llama-3.2 <sub>3B</sub>	<b>64.4</b>	<b>77.7</b>	<b>41.6</b>	<b>57.6</b>	<b>49.1</b>	<b>71.7</b>	<b>48.1</b>
Qwen-2.5 <sub>3B</sub>	62.6	<b>79.6</b>	41.4	<b>57.7</b>	48.7	71.5	<b>48.9</b>
<i>Impact of Different Optimization Algorithms</i>							
GRPO	<b>64.4</b>	<b>77.7</b>	<b>41.6</b>	<b>57.6</b>	<b>49.1</b>	<b>71.7</b>	<b>48.1</b>
PPO	58.7	68.9	37.3	53.7	47.3	70.4	45.5

Table 2: F1 scores (%) of the ablation study. Unless otherwise specified, Qwen-2.5-7B and Llama-3.2-3B are employed as the default generator and check model, respectively.

yields strong answer quality across diverse generation models. These findings indicate that the intermediate evidence extracted from R-Search’s reasoning-search trajectories encapsulates essential and reusable knowledge for accurate answer generation, and that such knowledge transfers effectively across models of different capacities. Leveraging this property, we reformulate the role of R-Search as R-Search-Tool, where it serves as a transferable evidence middleware (see Appendix C for the case study). Replacing standard retrieval in Vanilla RAG with R-Search-Tool yields consistent performance gains across models, and allows complex reasoning-search interactions to be optimized and carried out within small-scale models, rather than relying on powerful black-box models.

## 5.2 Ablation Study

In Table 2, we conduct ablation studies to analyze the effectiveness of the evidence, evidence-based check model, and optimization algorithm.

**Impact of Evidence.** We first evaluate the effec-

tiveness of the evidence mechanism. It consists of evidence inference and evidence-related rewards  $R^e$ , where  $R^e$  includes both the evidence reward  $r_\phi^e$  and the evidence formatting reward. “R-Search w/o Evidence” refers to a variant where the evidence is removed from rollout, and all evidence-related rewards  $R^e$  are disabled. We observe a clear performance drop without the evidence component and its associated rewards, confirming its importance. This drop is especially pronounced on complex multi-hop datasets compared to single-hop datasets. On the one hand, during the reasoning stage, multi-step search and long reasoning chains are prone to introducing irrelevant information. Evidence helps the LLM identify and extract the most relevant knowledge from a global perspective, preventing it from over-focusing on a single reasoning path and missing other valuable clues. On the other hand, evidence provides intermediate reward signals along the long reasoning-search trajectory, guiding the model to prioritize the reliability and completeness of intermediate factual content, rather than relying on speculative strategies that may yield unintended correct answers.

**Impact of Evidence Reward.** We evaluate the effectiveness of the evidence-related rewards  $R^e$ . The “w/o  $R^e$ ” setting removes  $R^e$  while retaining the evidence generation. Results show that preserving evidence inference without incorporating it into the optimization reward still leads to significant performance degradation. During multi-reward optimization, the quality of knowledge in the evidence is not leveraged to guide the optimization of the reasoning-search trajectory. As a result, relying solely on the answer reward fails to correct the quality of intermediate evidential content, potentially leading to deviations in the overall reasoning-search-evidence trajectory and performance decline. This finding highlights the critical role of the  $R^e$ .

**Impact of Different Evidence-Based Check Models.** Given the pivotal role of  $R^e$ , we conduct experiments across different types of LLMs to validate the generality of the evidence-related reward construction pipeline. The results show that various models can serve effectively as evidence-based checkers, consistently achieving strong performance. In particular, employing a model that differs from the policy model tends to yield more competitive results.

**Impact of Different Optimization Algorithms.** Both GRPO and PPO lead to strong results, which

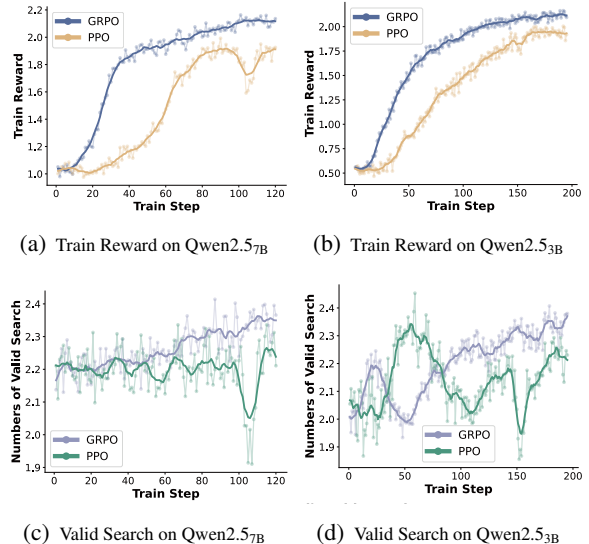


Figure 4: Train Reward and Valid Search.

highlights the generality of R-Search with mainstream RL algorithms. Furthermore, GRPO generally outperforms PPO, particularly on larger models and more complex multi-hop tasks. This performance gap may be attributed to differences in advantage estimation, as PPO relies on a learned critic to estimate the baseline, while GRPO uses the mean reward within a group as the baseline. This design may reduce the variance of policy gradient estimation, leading to more stable policy updates.

### 5.3 Effectiveness of Multi-Reward Optimization

In this experiment, we evaluate the effectiveness of multi-reward optimization under two RL algorithms: GRPO and PPO. As shown in Figure 4, the reward steadily increases as the number of training steps grows. This trend holds consistently across models of different scales and optimization strategies, demonstrating that the multi-reward optimization and reward modeling in R-Search are both effective and robust.

Furthermore, under the same number of training steps, GRPO yields a higher reward ceiling and faster convergence compared to PPO, leading GRPO-based R-Search to achieve stronger performance. This can be partly attributed to PPO’s reliance on an actor-critic architecture, where an insufficiently fitted value function in the early stage may increase estimation variance. In contrast, GRPO circumvents explicit value estimation, making it more likely to reach a higher reward ceiling.

## 5.4 Characteristics of Search in R-Search

In this experiment, we investigate the characteristics of search under multi-reward optimization. Figure 4 shows the learning dynamics and trends in the number of valid searches as training progresses. We observe that the trained models tend to trigger more search steps, engaging in more rounds of reasoning-search interaction and enabling deeper exploration of external knowledge. This sheds light on why R-Search yields more substantial improvements on complex multi-hop QA tasks.

## 6 Conclusion

We propose R-Search, an RL-based RAG framework, which interleaves multi-step reasoning and dynamic search within a unified decision process. By multi-reward optimization, R-Search enables robust policy learning that ensures both the soundness of intermediate reasoning and the completeness of retrieved knowledge. Extensive experiments demonstrate its superiority over strong RAG baselines. Moreover, we introduce R-Search-Tool, which modularizes distilled knowledge into transferable evidence components to enhance downstream answer generation for various models.

## Limitations

To demonstrate the strong generalization capability of R-Search, we only use the 2WikiMQA training dataset during the training phase. Although our method has achieved significant performance improvements on both in-domain and out-of-domain tasks under this setup, we acknowledge that incorporating more high-quality knowledge from diverse domains during training may further enhance the model’s performance. Therefore, future work should explore integrating more diverse and high-quality knowledge sources to further improve the effectiveness of R-Search.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272457) and the Youth Innovation Promotion Association of the Chinese Academy of Sciences (No. 2023171).

## References

Akari Asai and Hannaneh Hajishirzi. 2020. [Logic-guided data augmentation and regularization for consistent question answering](#). In *Proceedings of the*

*58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5642–5650. Association for Computational Linguistics.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Qinyuan Cheng, Xiaonan Li, Shimin Li, Qin Zhu, Zhangyue Yin, Yunfan Shao, Linyang Li, Tianxiang Sun, Hang Yan, and Xipeng Qiu. 2024. [Unified active retrieval for retrieval augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 17153–17166. Association for Computational Linguistics.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. 2023. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning*, pages 6948–6967. PMLR.

DeepSeek-AI. 2025. [Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.

Wenqi Fan, Yujian Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. [A survey on RAG meeting llms: Towards retrieval-augmented large language models](#). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6491–6501. ACM.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. [Deeprag: Thinking to retrieval step by step for large language models](#). *CoRR*, abs/2502.01142.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, 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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helvar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. [Openai o1 system card](#). *CoRR*, abs/2412.16720.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. [Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pages 7036–7050. Association for Computational Linguistics.
- Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. [Search-r1: Training llms to reason and leverage search engines with reinforcement learning](#). *CoRR*, abs/2503.09516.
- Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. [Flashrag: A modular toolkit for efficient retrieval-augmented generation research](#). In *Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025*, pages 737–740. ACM.
- Mandar Joshi, Eunsol Choi, Daniel S. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.
- Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. [Sure: Summarizing retrievals using answer candidates for open-domain QA of llms](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. 2025. [Training language models to self-correct via reinforcement learning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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](#). *Trans. Assoc. Comput. 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*.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. [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.
- 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)*, ACL 2023, Toronto, Canada, July 9-14, 2023, pages 9802–9822. Association for Computational Linguistics.
- Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

- Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. 2024. [Rule based rewards for language model safety](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 108877–108901. Curran Associates, Inc.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Bäck. 2026. [Multi-step reasoning with large language models, a survey](#). *ACM Comput. Surv.*, 58(6):160:1–160:35.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-context retrieval-augmented language models](#). *Trans. Assoc. Comput. Linguistics*, 11:1316–1331.
- Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021. [End-to-end training of multi-document reader and retriever for open-domain question answering](#). In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25968–25981.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *CoRR*, abs/1707.06347.
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. [Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- 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](#). *CoRR*, abs/2503.05592.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Musique: Multi-hop questions via single-hop question composition](#). *Trans. Assoc. Comput. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.
- Ruobing Wang, Qingfei Zhao, Yukun Yan, Daren Zha, Yuxuan Chen, Shi Yu, Zhenghao Liu, Yixuan Wang, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. [Deepnote: Note-centric deep retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 19688–19715. Association for Computational Linguistics.
- Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. [Large language models are better reasoners with self-verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 2550–2575. Association for Computational Linguistics.
- Marco A Wiering and Martijn Van Otterlo. 2012. Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729.
- 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, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.
- Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. [Making retrieval-augmented language models robust to irrelevant context](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun,

Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, and 36 others. 2024. [Chatglm: A family of large language models from GLM-130B to GLM-4 all tools](#). *CoRR*, abs/2406.12793.

Jixiao Zhang and Chunsheng Zuo. 2025. [GRPO-LEAD: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 5642–5654. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *CoRR*, abs/2309.01219.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024. [Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22600–22632. Association for Computational Linguistics.

## A Appendix A: Experiment and Result

### A.1 Datasets

We conduct extensive experiments on seven English datasets, covering both complex multi-hop and simpler single-hop QA tasks. The multi-hop QA task serves to evaluate whether R-Search can handle complex logic- and knowledge-intensive questions. The single-hop QA task assesses its ability to address knowledge-intensive questions and explore its robustness across questions with varying levels of complexity.

For the multi-hop task, we adopt four challenging multi-hop QA datasets: **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA (2WikiMQA)** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022), and **Bamboogle** (Press et al., 2023). The datasets require models to start from the original query and explore a logical, knowledge-driven multi-hop reasoning path to answer each intermediate sub-query and reach the final answer. We follow the train and test splits released by (Trivedi et al., 2023) for the first three datasets, each with 500 test samples. Bamboogle evaluation uses all 125 test samples.

For the single-hop task, we select three factoid-based QA datasets, including **NQ** (Kwiatkowski et al., 2019), **PopQA** (Mallen et al., 2023), and **TriviaQA** (Joshi et al., 2017). These datasets require models to collect specific passages and identify key factual information to answer the questions correctly. We use the test sets and randomly sample 500 examples from each dataset for evaluation.

### A.2 More Results

We present the results of all evaluation metrics for the ablation study in Table 3, Table 4 and Table 5. The trends are consistent with the main results in the ablation study analysis.

### A.3 More Implementation Details

During inference, we set the temperature to 0.1 across all models to reduce uncertainty. For a fair comparison, we align the training hyperparameters and datasets of Search-R1 with those used in our method. For other baseline methods, we evaluate their performance using FlashRAG (Jin et al., 2025b).

## B Appendix B: Template

### B.1 Rollout Template

We design a system template for the rollout phase to guide the model through the complete interac-

You are a helpful assistant that can solve the given question step by step. For each step, start by explaining your thought process. If additional information is needed, provide a specific query enclosed in `<search>` and `</search>`. The system will return the top search results within `<observation>` and `</observation>`. You can perform multiple searches as needed. When you know the final answer, use `<original_evidence>` and `</original_evidence>` to provide all potentially relevant original information from the observations. Ensure the information is complete and preserves the original wording without modification. If no searches were conducted or observations were made, omit the evidence section. Finally, provide the final answer within `<answer>` and `</answer>` tags.

Figure 5: Rollout Template. The question is appended at the end during training and inference.

tion process, from receiving the input question  $q$  to generating the final answer  $\alpha$ . This template covers four key stages: reasoning, search, evidence integration, and answer generation. Notably, for questions that do not require external knowledge, the LLM automatically determines that search is unnecessary. In such cases, the template instructs the model to perform only reasoning and answer generation. During training, we initialize the training process with the system template and the user’s question. It is worth noting that we do not impose a manual separation between reasoning and retrieval (e.g., by encapsulating reasoning within a `<think>` and `</think>` tag); Instead, we provide a high-level instruction that guides the model to explain its thought process before each action, without imposing any specific format for reasoning text. This stems from the fact that LLM generation is inherently a form of reasoning, where every generated token can be viewed as part of a thought chain for question-solving. For parts such as `<search>`, `<observation>`, `<original_evidence>` and `<answer>`, which are distinct from general thought reasoning, we introduce specific token tags to mark them, allowing the LLM to recognize the boundaries between different functional segments.

### B.2 Evidence Template

Figure 6 presents the instruction template used for generating evidence during rollout. Before outputting the final answer  $\alpha$ , we instruct the LLM to generate evidence based on the original question  $q$  and all previously retrieved texts (provided in

```
Answer the question based on the given passages.  
Only give me the answer and do not output any  
other words. The following are given passages:  
{evidence}  
Question: {query}  
Answer:
```

Figure 6: Evidence Template.

the `<observation>` and `</observation>`). This evidence helps the LLM rethink the retrieved information from a global perspective and focus on key factual knowledge. By seamlessly integrating the evidence into the reasoning process, we facilitate a deeper interaction between reasoning and retrieved knowledge.

### **C Appendix C: Case Study**

Tables 6 and 7 show examples of the reasoning process from RL-trained models on 2-hop and more complex 4-hop questions. In these cases, R-Search uses multi-type rewards to improve the reasoning-search process. The model usually starts by generating a general reasoning plan, retrieves information when needed, and uses intermediate conclusions to guide the next search action. The evidence generated through this interaction is clear, well-structured, and informative, making it easy to transfer to downstream models for final answer generation.

Method	Multi-Hop QA									Single-Hop QA						
	2WikiMQA		HotpotQA		MuSiQue		Bamboogle			NQ		TriviaQA		PopQA		
	EM	F1	EM	F1	EM	F1	EM	F1	Avg.	EM	F1	EM	F1	EM	F1	Avg.
	<i>Qwen-2.5-3B</i>															
R-Search	65.0	72.6	43.4	54.4	25.8	34.8	37.6	49.8	47.9	35.2	46.0	56.0	64.0	37.0	44.9	47.2
R-Search w/o $R^e$	62.2	70.2	43.0	53.8	20.6	28.3	39.2	49.5	45.9	34.0	44.1	54.4	61.7	34.8	42.7	45.3
R-Search w/o Evidence	58.2	66.6	41.0	53.4	23.6	33.3	32.0	42.3	43.8	34.6	44.6	54.4	62.3	36.0	41.6	45.6
	<i>Qwen-2.5-7B</i>															
R-Search	69.8	77.7	52.2	64.4	31.4	41.6	42.4	57.6	54.6	38.0	49.1	64.2	71.7	41.8	48.1	52.1
R-Search w/o $R^e$	69.2	76.5	42.4	53.7	24.6	35.6	38.4	50.0	48.8	37.8	48.2	60.2	66.6	40.8	46.1	50.0
R-Search w/o Evidence	69.0	77.5	49.4	61.9	29.4	39.6	44.0	55.9	53.3	36.8	47.3	63.6	70.2	41.4	48.0	51.2

Table 3: Impact of evidence (%).

Models	Multi-Hop QA									Single-Hop QA						
	2WikiMQA		HotpotQA		MuSiQue		Bamboogle			NQ		TriviaQA		PopQA		
	EM	F1	EM	F1	EM	F1	EM	F1	Avg.	EM	F1	EM	F1	EM	F1	Avg.
	<i>Qwen-2.5-3B</i>															
Llama-3.2 <sub>3B</sub>	65.0	72.6	43.4	54.4	25.8	34.8	37.6	49.8	47.9	35.2	46.0	56.0	64.0	37.0	44.9	47.2
Qwen-2.5 <sub>3B</sub>	62.8	70.8	47.0	58.6	25.4	34.6	36.8	49.2	48.1	34.6	45.3	56.2	63.8	37.2	44.3	46.9
	<i>Qwen-2.5-7B</i>															
Llama-3.2 <sub>3B</sub>	69.8	77.7	52.2	64.4	31.4	41.6	42.4	57.6	54.6	38.0	49.1	64.2	71.7	41.8	48.1	52.1
Qwen-2.5 <sub>3B</sub>	72.0	79.6	49.8	62.6	30.0	41.4	43.2	57.7	54.5	37.2	48.7	63.4	71.5	42.2	48.9	52.0

Table 4: Impact of different evidence-based check models (%).

Algorithm	Multi-Hop QA									Single-Hop QA						
	2WikiMQA		HotpotQA		MuSiQue		Bamboogle			NQ		TriviaQA		PopQA		
	EM	F1	EM	F1	EM	F1	EM	F1	Avg.	EM	F1	EM	F1	EM	F1	Avg.
	<i>Qwen-2.5-3B</i>															
GRPO	65.0	72.6	43.4	54.4	25.8	34.8	37.6	49.8	47.9	35.2	46.0	56.0	64.0	37.0	44.9	47.2
PPO	50.8	60.3	42.0	52.4	24.0	33.5	37.6	50.0	43.8	34.0	43.7	54.6	63.1	36.8	44.2	46.1
	<i>Qwen-2.5-7B</i>															
GRPO	69.8	77.7	52.2	64.4	31.4	41.6	42.4	57.6	54.6	38.0	49.1	64.2	71.7	41.8	48.1	52.1
PPO	59.8	68.9	47.0	58.7	27.0	37.3	40.0	53.7	49.1	37.0	47.3	63.2	70.4	37.8	45.5	50.2

Table 5: Impact of different optimization algorithms (%).

---

---

**Question:** When was countrywide bought by the company that bought FleetBoston Financial?

---

"Thought": To answer this question, we need to follow these steps:

1. Identify the company that bought FleetBoston Financial.
2. Determine when this company bought Countrywide.

Let's start with step 1.

<search> FleetBoston Financial was bought by whom? </search>

<observation> (Title: "Bank of America") In 2004, Bank of America announced it would purchase Boston - based bank FleetBoston Financial for \$47 billion in cash and stock. By merging with Bank of America, all of its banks and branches were given the Bank of America logo. At the time of merger, FleetBoston was the seventh largest bank in United States with \$197 billion in assets, over 20 million customers and revenue of \$12 billion.

(Title: "Boston") Boston is an intellectual, technological, and political center but has lost some important regional institutions . . .

(Title: "Boston") Other important industries are financial services . . .

(Title: "The Ritz-Carlton Hotel Company") Blackstone Real Estate Acquisitions of New York bought The Ritz . . .

(Title: "RBC Bank") On June 19, 2011, PNC Financial Services agreed to buy RBC Bank USA from Royal Bank of Canada for \$3.45 billion . . . <observation>

From the observations, we can conclude that FleetBoston Financial was bought by Bank of America in 2004.

Now, let's move to step 2.

<search> When did Bank of America buy Countrywide? </search>

<observation> (Title: "Bank of America Home Loans") On January 11, 2008, Bank of America announced that it planned to purchase Countrywide Financial for \$4.1 billion in stock. On June 5, 2008, Bank of America Corporation announced it had received approval from the Board of Governors of the Federal Reserve System to purchase Countrywide Financial Corporation. Then, on June 25, 2008, Countrywide announced it had received the approval of 69% of its shareholders to the planned merger with Bank of America. On July 1, 2008, Bank of America Corporation completed its purchase of Countrywide Financial Corporation. In 1997, Countrywide had spun off Countrywide Mortgage Investment as an independent company called IndyMac Bank. Federal regulators seized IndyMac on July 11, 2008, after a week - long bank run . . . <observation>

(Title: "Bank of America Home Loans") Bank of America Home Loans is the mortgage unit of Bank of America. . .

(Title: "Bank of America") In 2004, Bank of America announced it would purchase Boston - based bank FleetBoston Financial for \$47 billion in cash and stock. . .

(Title: "Bank of America") Its acquisition of investment bank Merrill Lynch in 2008 solidified it as a "Bulge Bracket bank" . . .

(Title: "General Tire") GenCorp sold General Tire to German tire maker Continental AG in 1987 . . .

From the observations, we can conclude that Bank of America bought Countrywide on July 1, 2008.

<original\_evidence>- FleetBoston Financial was bought by Bank of America in 2004.

- Bank of America bought Countrywide on July 1, 2008. <original\_evidence>

<answer> July 1, 2008 </answer> "✓"

---

---

Table 6: An 2-hop example of R-Search on MuSiQue dataset using PPO. Orange segments denote those used to support correct answers. The "✓" indicates a correct answer.

---

---

**Question:** Which film whose director is younger, My Baby’S Daddy or A Tale Of Winter?

"Thought": " To determine which film has a director who is younger, we need to identify the directors of both films and compare their ages.

First, let’s identify the director of "My Baby’s Daddy".

<search> Who directed My Baby’s Daddy? </search>

<observation> (Title: "My Baby’s Daddy") My Baby’s Daddy is a 2004 American comedy film, directed by Cheryl Dunye.

(Title: "Alec Mapa: Baby Daddy") Alec Mapa: Baby Daddy is a 2013 one- man show by Alec Mapa . . .

(Title: "Bachelor Daddy") Bachelor Daddy is a 1941 American comedy film directed by Harold Young and written by Robert Lees and Frederic I. Rinaldo . . .

(Title: "Baby and I") Baby and I is a 2008 South Korean film about a rebellious 18- year- old high school senior who ends up caring for a baby with a note . . .

(Title: "Who’s Your Daddy? (film)") Who’s Your Daddy? is a 2002 American comedy film directed( and co-scripted) by Andy Fickman. <observation>

The director of "My Baby’s Daddy" is Cheryl Dunye.

Next, let’s identify the director of "A Tale Of Winter".

<search> Who directed A Tale Of Winter? </search>

<observation> (Title: "A Tale of Winter") A Tale of Winter is a 1992 French drama film directed by Eric Rohmer, and starring Charlotte Very, Frederic van den Driessche and Michael Voletti. It is the second of Rohmer’s "Tales of the Four SeasonsContes des quatre saisons"), which also include "A Tale of Springtime" (1990), "A Summer’s Tale" (1996) and "Autumn Tale" (1998) . . .

(Title: "A Tale of Winter") A Tale of Winter is a 1992 French drama film directed by Eric Rohmer, and starring Charlotte Very, Frederic van den Driessche and Michael Voletti. It is the second of Rohmer’s" Tales of the Four SeasonsContes des quatre saisons") . . .

. . . <observation>

The director of "A Tale of Winter" is Eric Rohmer.

Now, we need to determine the ages of Cheryl Dunye and Eric Rohmer to compare.

<search> When was Cheryl Dunye born? </search>

<observation> (Title: "Cheryl Dunye") Cheryl Dunye( born May 13, 1966) is a Liberian- American film director, producer, screenwriter, editor and actress. Dunye’s work often concerns themes of race, sexuality, and gender, particularly issues relating to black lesbians. . . .

. . .

Cheryl Dunye was born on May 13, 1966.

<search> When was Eric Rohmer born? </search>

<observation> (Title: "Eric Rohmer") Jean Marie Maurice Scherer or Maurice Henri Joseph Scherer, known as Eric Rohmer (21 March 192011 January 2010), was a French film director, film critic, journalist, novelist, screenwriter, and teacher. Rohmer was the last of the post-World War II French New Wave directors to become established. He edited the influential film journal, "Cahiers du cinema", from 1957 to 1963, while most of his . . .

(Title: "Eric Rohmer filmography") This is a list of films by the French director Eric Rohmer.

. . . <observation>

Eric Rohmer was born on March 21, 1920.

Now we compare the birth years:

- Cheryl Dunye: 1966

- Eric Rohmer: 1920

Cheryl Dunye is younger than Eric Rohmer.

Therefore, the film "My Baby’s Daddy" has a younger director.

<original\_evidence>- "My Baby’s Daddy" was directed by Cheryl Dunye, born in 1966.

- "A Tale of Winter" was directed by Eric Rohmer, born in 1920.

- Cheryl Dunye is younger than Eric Rohmer. <original\_evidence>

<answer> My Baby’S Daddy </answer> "✓"

---

---

Table 7: An 4-hop example of R-Search on 2WikiMQA dataset using GRPO. Orange segments denote those used to support correct answers. The "✓" indicates a correct answer.