Process vs. Outcome Reward: Which is Better for Agentic RAG Reinforcement Learning

Wenlin Zhang[1,*], **Xiangyang Li**[2,*], **Kuicai Dong**[2,*], **Yichao Wang**[2,†], **Pengyue Jia**[1], **Xiaopeng Li**[1], **Yingyi Zhang**[1], **Derong Xu**[1], **Zhaocheng Du**[2], **Huifeng Guo**[2], **Ruiming Tang**[2], **Xiangyu Zhao**[1,†]
[1] City University of Hong Kong, Hong Kong, [2] Noah's Ark Lab, Huawei, China
{wl.z, jia.pengyue, xiaopli2-c, yzhang6375-c, derongxu2-c}@my.cityu.edu.hk,
{lixiangyang34, dong.kuicai, wangyichao5, zhaochengdu, huifeng.guo, tangruiming}@huawei.com,
xianzhao@cityu.edu.hk

Abstract

Retrieval-augmented generation (RAG) enhances the text generation capabilities of large language models (LLMs) by integrating external knowledge and up-to-date information. However, traditional RAG systems are limited by static workflows and lack the adaptability required for multistep reasoning and complex task management. To address these limitations, agentic RAG systems (e.g., DeepResearch) have been proposed, enabling dynamic retrieval strategies, iterative context refinement, and adaptive workflows for handling complex search queries beyond the capabilities of conventional RAG. Recent advances, such as Search-R1, have demonstrated promising gains using outcome-based reinforcement learning, where the correctness of the final answer serves as the reward signal. Nevertheless, such outcome-supervised agentic RAG methods face challenges including low exploration efficiency, gradient conflict, and sparse reward signals. To overcome these challenges, we propose to utilize fine-grained, process-level rewards to improve training stability, reduce computational costs, and enhance efficiency. Specifically, we introduce a novel method `ReasonRAG` that automatically constructs `RAG-ProGuide`, a high-quality dataset providing process-level rewards for (i) query generation, (ii) evidence extraction, and (iii) answer generation, thereby enhancing model inherent capabilities via process-supervised reinforcement learning. With the process-level policy optimization, the proposed framework empowers LLMs to autonomously invoke search, generate queries, extract relevant evidence, and produce final answers. Compared to existing approaches such as Search-R1 and traditional RAG systems, `ReasonRAG`, leveraging `RAG-ProGuide`, achieves superior performance on five benchmark datasets using only 5k training instances, significantly fewer than the 90k training instances required by Search-R1. Our code is available at https://github.com/Applied-Machine-Learning-Lab/ReasonRAG.

1 Introduction

Large language models (LLMs) [1, 2, 3] have demonstrated substantial proficiency in text generation and natural language understanding [4, 5], revealing their potential for powering various downstream applications such as recommender systems [6]. However, their reliance on static training data constrains their ability to address dynamic and real-time queries, often resulting in outdated or hallucinated information [7, 8, 9]. Retrieval-Augmented Generation (RAG) [10] has emerged as a

[*]Equal contribution
[†]Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-1965

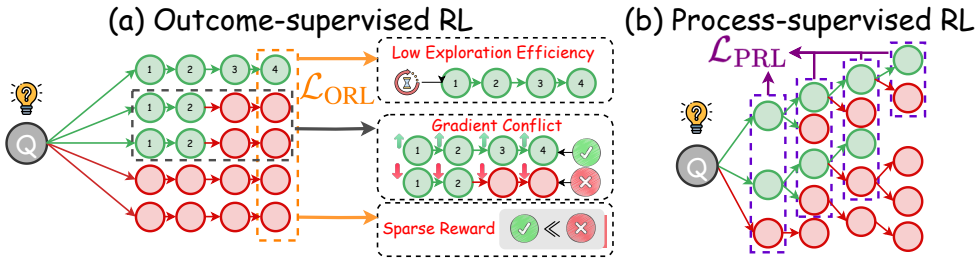


Figure 1: Outcome-supervised vs. process-supervised RL for multi-step reasoning. Each circle denotes one-step reasoning response, where the correct response is colored as "green" and error response is colored as "red".

promising solution by equipping LLMs with external knowledge sources, improving the relevance, factual accuracy, and timeliness of responses [11, 12, 13]. Despite these advances, traditional RAG architectures are limited by their linear and static workflows, which suffer from complex multi-step reasoning, deep contextual integration, and iterative response refinement [10, 14]. To address these shortcomings, agentic RAG (*e.g.,* DeepResearch [15, 16, 17]) systems have been developed, enabling dynamic retrieval strategies, enhanced contextual understanding, and iterative refinement. Achieving agentic RAG requires the underlying LLMs to orchestrate retrieval, filter relevant information, and iteratively refine their outputs, resulting in more adaptive and efficient information processing.

To advance agentic RAG, early approaches [18, 19, 20, 21, 22] primarily focused on prompt engineering to adapt powerful LLMs to agentic workflows. However, due to the limited reasoning and instruction-following capabilities of LLMs, supervised fine-tuning (SFT) methods [23] have been introduced, extending prompt-based approaches by directly optimizing and refining model parameters. Due to SFT storing reasoning steps within the model parameters, the improved reasoning capabilities often encounter challenges in generalizing across different domains [24]. More recently, reinforcement learning (RL) methods (*e.g.,* OpenAI-O1 [25] and DeepSeek-R1 [26] achieve notable improvements in LLM reasoning by employing outcome-supervised RL techniques. Building on these developments, Search-R1 [27] incorporates a search engine as part of the LLM's environment and leverages outcome-based reinforcement learning, using the correctness of the final answer as the reward signal. These advances demonstrate that outcome-supervised reinforcement learning can substantially enhance the capabilities required for agentic RAG, enabling straightforward, end-to-end optimization of the entire workflow.

Despite its promise, outcome-supervised RL also presents inherent limitations, as illustrated in Figure 1. First, **low exploration efficiency** occurs since the model must generate a complete reasoning chain before receiving any reward [27]. Ideally, the reward should be given when errors occur at intermediate steps to facilitate learning. Second, **gradient conflict** arises when mistakes occur late in the reasoning process; the entire sequence (including correct early steps) is penalized [28]. This can lead to conflicting gradients that can push correct actions in the wrong direction. Third, **the rewards are sparse**, as outcome-supervised RL only provides feedback upon producing the final answer [29]. Reward sparsity relies on more training data and steps to converge, as the model receives infrequent learning signals. In contrast, **process-supervised RL** addresses these issues by providing fine-grained, stepwise rewards throughout the reasoning process, enabling more efficient exploration, reducing gradient conflict, and accelerating model learning through denser feedback.

However, applying process-supervised RL to RAG presents several key challenges. (1) **Process Reward Design**: Effective process rewards are essential for guiding the model toward the shortest and most efficient correct reasoning path. Rewards must incentivize helpful intermediate steps that lead to the correct final answer, while penalizing unnecessarily long or circuitous reasoning sequences [30]. (2) **Exploration Efficiency and Annotation Cost**: While human annotators who are skilled in information retrieval can create high-quality process-level annotations by decomposing complex retrieval tasks into efficient steps, this approach is prohibitively expensive due to the substantial manual effort involved [31]. In contrast, autonomous RAG agents can generate a wide range of possible retrieval and reasoning steps, but this large search space makes it difficult to identify and select high-quality, meaningful steps for use as process-level annotations.

https://doi.org/10.52202/085713-1965

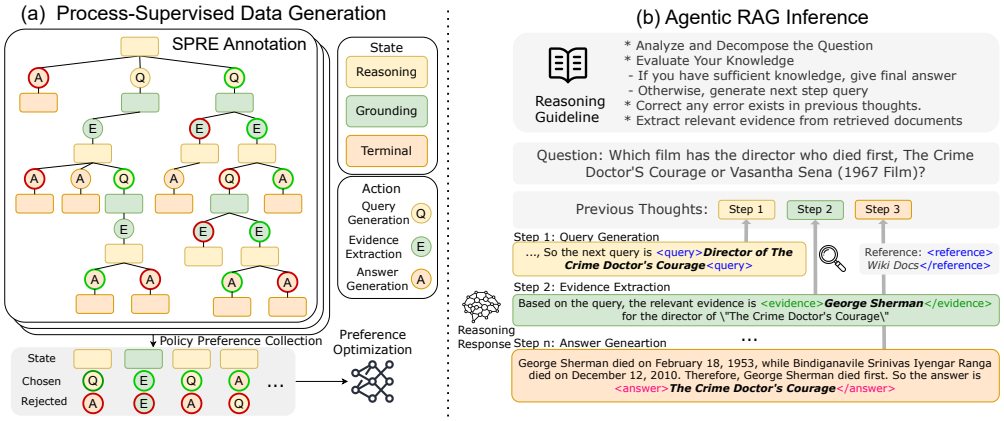


Figure 2: Framework of ReasonRAG. Figure (a) illustrates the policy optimization based on process supervision. MCTS guides the construction of the state-action tree and the assignment of process-level rewards for fine-grained policy optimization. (Actions derived from the same state are color-coded by reward: green circle (highest), red circle (lowest).) Figure (b) demonstrates an inference example.

To address these challenges, we propose ReasonRAG, an advanced process-supervised RL method to enhance agentic RAG reasoning. Specifically, ReasonRAG employs Monte Carlo Tree Search (MCTS) [32] as a search strategy to efficiently balance exploration and exploitation, enabling thorough exploration of diverse reasoning paths and identification of high-reward intermediate steps for guiding the RAG process. Building on these paths, we introduce a novel **S**hortest **P**ath **R**eward **E**stimation (SPRE) algorithm to assign rewards. SPRE favors sequences that lead to the correct answer while penalizing unnecessarily lengthy reasoning, thereby promoting efficiency. This approach yields RAG-ProGuide, a dataset comprising 5k queries with 13,000 high-quality process-level preference pairs. Using RAG-ProGuide and our process-supervised Direct Preference Optimization (DPO) [33] strategy, ReasonRAG is further trained to make dynamic decisions, such as whether to invoke retrieval, formulate subsequent search queries, analyze retrieved documents for relevant evidence, and synthesize evidence into final answers. Extensive experiments on five benchmark RAG datasets show ReasonRAG (trained with only 13k process-level steps) outperforms Search-R1 (trained on 90k queries with approximately 270k intermediate steps), suggesting the superiority of process-supervised RL over outcome-supervised RL. Our key contributions can be summarized as follows:

- We propose ReasonRAG, an automatic framework for agentic RAG process-level reward annotations. We introduce SPRE for efficient RAG process-level reward annotation and MCTS for high-quality decision space exploration.
- We introduce a process-level annotation dataset RAG-ProGuide, which serves as an off-policy dataset, and can be easily applied for various LLM policy optimization.
- We conduct extensive comparative experiments of outcome-supervised RL and process-supervised RL for RAG reasoning with Qwen2.5-7B-Instruct. The experimental results on five benchmark datasets demonstrate the superiority and training efficiency of ReasonRAG.

2 ReasonRAG Framework

2.1 Framework Overview

This section details the design of ReasonRAG framework, as depicted in Figure 2. Figure 2a outlines our approach for constructing high-quality process-supervised data. We first introduce *Shortest Path Reward Estimation* (SPRE) to provide process-level supervision reward (see Section 2.2.1). To efficiently gather these rewards, we employ Monte Carlo Tree Search (MCTS) algorithm to explore the vast decision space in agentic RAG and collect informative intermediate steps (see Section 2.2.2).

The resulting process-supervised dataset, RAG-ProGuide (see Section 2.2.3), is then used to optimize ReasonRAG via *policy preference optimization*. This training strategy guides the model to prefer desirable reasoning trajectories in agentic RAG (see Section 2.3).

https://doi.org/10.52202/085713-1965

Figure 2b illustrates the agentic RAG inference workflow in `ReasonRAG`. During inference, the model adaptively conducts reasoning by dynamically invoking search engine and interleaving three core actions: query generation, evidence extraction, and answer generation (see Section 2.4).

2.2 Process-Supervised Data Generation

Effective process-supervised policy optimization requires high-quality, granular reward signals at the process level. As outlined in Section 1, generating such rewards for agentic RAG presents two main challenges: (1) the absence of reward functions for intermediate reasoning steps, and (2) the lack of an efficient and cost-effective method to generate informative reasoning trajectories. To overcome these challenges, we introduce a novel process-level reward function, SPRE (see Section 2.2.1), specifically designed for agentic RAG. Furthermore, we develop an MCTS-based approach (see Section 2.2.2) to efficiently explore the decision space and collect high-quality process-level data.

2.2.1 Shortest Path Reward Estimation (SPRE)

Unlike outcome-level rewards, process-level rewards provide supervision at each intermediate step of agentic RAG. A key challenge is the absence of ground-truth reward signals for partial reasoning trajectories. Furthermore, due to the large decision space, the reward function must account for both final correctness and reasoning efficiency. To address these challenges, we propose *Shortest Path Reward Estimation (SPRE)*, which evaluates the quality of each intermediate reasoning path by simulating its possible outcomes and penalizing unnecessarily long trajectories.

Formally, the agentic RAG process consists of an n-step sequence $[y_1, \cdots, y_n]$, where each y_i represents the output of a single reasoning step, conditioned on the initial question x and previous steps $y_{<i}$. To evaluate a partial sequence $y_{1:t}$, we simulate multiple continuations, known as *rollouts*, until a final answer is obtained. By repeating the rollout process k times and scoring each outcome, we approximate the reward as a Monte Carlo-style estimation with step-based penalties:

$$Q_t = \text{MonteCarlo}(x, y_{1:t}) = \frac{1}{k} \sum_{i=1}^{k} v(\text{rollout}_i) \cdot \alpha^{\text{step}(\text{rollout}_i)} \tag{1}$$

Here, rollout_i is the i-th simulated completion of $y_{1:t}$, $v(\text{rollout}_i) \in [0, 1]$ denotes the correctness score (*e.g.,* F_1 match to the ground truth), and $\text{step}(\text{rollout}_i)$ is the number of total reasoning steps in the trajectory. The decay factor $\alpha \in (0, 1]$ penalizes unnecessarily long reasoning paths. This reward encourages the model to favor trajectories that achieve correct answers with fewer steps, thus balancing accuracy and efficiency in agentic RAG reasoning.

2.2.2 Monte Carlo Tree Search (MCTS) for Process-level Exploration

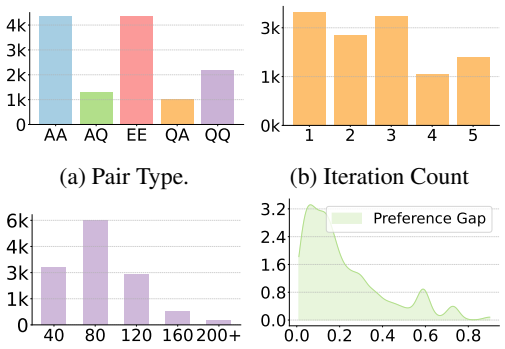
Although SPRE offers reliable reward signals for evaluating intermediate steps, generating diverse yet meaningful trajectories remains challenging. The search space in agentic RAG is extensive due to open-ended nature of retrieval, which requires continuous refinement of search queries for relevant information. To address this, we propose a tailored MCTS framework for agentic RAG. MCTS enables efficient exploration by selectively expanding the most promising reasoning paths based on estimated rewards.

We adapt MCTS to agentic RAG context by explicitly defining states and actions for tree construction. Formally, each intermediate reasoning step is represented as a state $s = (x, y_{<i}, \text{stage})$, where x is the original question, $y_{<i}$ is the sequence of prior reasoning outputs, and stage $\in \{\text{Reasoning}, \text{Grounding}, \text{Terminal}\}$ indicates current point of agentic flow. Actions $\in \{\text{Query Generation}, \text{Evidence Extraction}, \text{Answer Generation}\}$ are determined by the current stage as follows:

- **Reasoning** stage: Choose between generating a new query for document retrieval or directly generating an answer. If a new query is generated, a retrieval operation is performed, and the retrieved documents are appended to the state for subsequent decisions. If an answer is produced, the process transitions to the *Terminal* stage.
- **Grounding** stage: Select evidence spans from the retrieved documents. Afterwards, the system returns to the *Reasoning* stage for further reasoning or answering.
- **Terminal** stage: End the exploration process when the final answer has been generated.

Statistics	Number
Questions	4603
- PopQA	704 (15.3%)
- HotpotQA	2843 (61.8%)
- 2WikiMultihopQA	1056 (22.9%)
Actions	13289
- Query Generation	3295 (24.8%)
- Evidence Extraction	4305 (32.4%)
- Answer Generation	5689 (42.8%)
Avg./Min./Med./Max. Iteration	2.7/1/3/5
Avg./Min./Med./Max. Tokens	65.5/9/60/625

Table 1: Overall Dataset Statistics



(a) Pair Type. (b) Iteration Count

(c) Token Length (d) Reward Gap

Figure 3: Dataset Distribution.

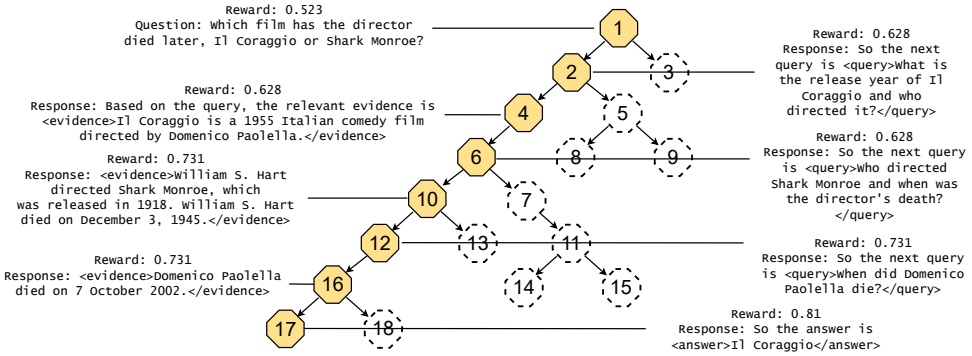


Figure 4: A Tree Example for the process-level preference annotations.

Based on the current state s, the policy for generating the next action a is defined as:

$$\pi(a \mid s) = \text{LLM}(a \mid s) = \begin{cases} \pi_\theta(\cdot \mid x, y_{<i}, p_{\text{stage}}), & \text{if stage is Reasoning} \\ \pi_\theta(\cdot \mid x, y_{<i}, \text{docs}, p_{\text{stage}}), & \text{otherwise} \end{cases} \tag{2}$$

State transitions are defined as $s_{t+1} = \text{concatenate}(s_t, a_t)$, where each action leads to a new node in the search tree, representing an extended reasoning sequence. This recursive process incrementally builds a tree rooted at the original question. MCTS then operates iteratively, performing three core steps: *selection*, *expansion*, and *backpropagation*. Specifically, at each iteration, MCTS selects promising paths using a Upper Confidence Bound (UCB) based objective, expands new states by sampling LLM-generated actions, and backpropagates SPRE-estimated rewards (see Equation 1) to update the tree (see more comprehensive explanation of MCTS exploration in Appendix A). Integrating MCTS with SPRE enables efficient exploration and prioritization of high-reward reasoning steps, producing high-quality process-level annotations to optimize the agentic RAG policy.

2.2.3 `RAG-ProGuide` Dataset

Construction. Using the MCTS-based exploration framework, we construct a high-quality process-supervised dataset `RAG-ProGuide` to facilitate process-level policy optimization. We randomly sample 3,000 questions each from PopQA [34], HotpotQA [35], and 2WikiMultihopQA [36], covering both single-hop and multi-hop question answering tasks. An advanced large language model serves as the policy model to simulate the agentic RAG reasoning process within the MCTS framework (see Section 2.2.2). During tree search, we prune all branches that do not yield a final answer. For each complete trajectory, we compute the F_1 score between the predicted answer and ground truth, and use this correctness signal to estimate intermediate node rewards via SPRE (see Section 2.2.1). These rewards are propagated through the MCTS tree to guide preference pair selection. To ensure high-quality preference data, we perform post-processing to remove duplicates and uninformative

https://doi.org/10.52202/085713-1965

comparisons: (i) we discard preference pairs with identical response sequences, and (ii) pairs with a reward difference less than 0.01. After filtering, the final dataset consists of 4,603 questions and 13,289 distinct preference pairs. Figure 4 illustrates an example for the tree-structured process data. The root node corresponding to the original question, each node corresponds one-step response from LLMs. The correct reasoning path has been colored in orange and annotated with a higher reward.

Dataset Statistics. Table 1 presents detailed statistics and distributions for our dataset. The questions are drawn from PopQA, HotpotQA, and 2WikiMultihopQA, providing comprehensive coverage of both single-hop and multi-hop reasoning within the RAG decision space. The dataset contains a balanced distribution of three reasoning actions, reflecting the multi-stage nature of the agentic RAG process. As shown in Figure 2a, the distribution of preference pair types demonstrates diverse comparative scenarios; the x-axis abbreviations (**A**: answer generation, **Q**: query generation, **E**: evidence extraction) indicate action types in accepted versus rejected paths. This diversity ensures fine-grained comparative coverage across different reasoning stages. Figure 2b indicates a range of reasoning iteration counts, consistent with the complexity of multi-hop inference. Figure 2c shows a broad distribution of response token lengths, confirming the dataset's capacity to capture various response complexities. Additionally, Figure 2d illustrates the probability distribution of the reward gap between preference pairs. The comprehensive coverage of this distribution across a wide range of values ensures that the preference learning is guided by a rich and informative signal. Collectively, these statistics demonstrate the dataset's quality and its suitability for training robust process-level decision policies in agentic RAG frameworks.

2.3 Process-Supervised Preference Optimization

Based on the process-supervised preference data, we apply DPO [33] to tune the policy model. The optimization objective can be denoted as follows:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y_{<t},y_t^w,y_t^l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{p_\theta(y_t^w|x,y_{<t})}{p_\theta(y_t^l|x,y_{<t})}\right)\right] \tag{3}$$

where x denotes the original question, $y_{<t}$ denote the the responses from previous reasoning steps, y_t^w and y_t^l represent the preferred and dispreferred responses in the subsequent step, respectively, and the hyperparameter β controls the KL constraint.

2.4 Agentic RAG Inference

To enable LLMs to autonomously interact with external information, we propose an agentic RAG workflow that supports adaptive reasoning through iterative search and reflection. `ReasonRAG` allows the model to dynamically determine when and how to invoke search engine based on question complexity.

The workflow operates through three recursive decision states: Reasoning, Grounding, and Terminal. In the **Reasoning** state, the LLM evaluates the current context to decide if it has sufficient information to answer the question. If sufficient, it generates a final answer enclosed in placeholders (`<answer>` and `</answer>`), thus terminating the process. If not, the model creates a new query enclosed in `<query>` and `</query>` placeholders to retrieve additional evidence. The system then transitions to the **Grounding** state, where documents are retrieved based on the query, and the model extracts relevant evidence spans. These evidence spans are appended to the context, after which the process loops back to the **Reasoning** state for further deliberation. (See Appendix F for detailed prompt designs.)

Algorithm 1 Agentic RAG Inference Pipeline

Require: Original question x, large language model π_θ, retriever $\mathcal{R}$, maximum reasoning round N.
Ensure: Final response y.

Initialize reasoning count $i \leftarrow 0$, and stage $\leftarrow$ Reasoning
for $i \leftarrow 0$ **to** $N-1$ **do**
  if stage is Reasoning **then**
    $y_i \sim \pi_\theta(\cdot|x, y_{<i}, p_{\text{stage}})$
  else
    $y_i \sim \pi_\theta(\cdot|x, y_{<i}, docs, p_{\text{stage}})$
  end if
  $y \leftarrow y + y_i$
  if <query> detected in y_i **then**
    stage $\leftarrow$ Grounding
    $q \leftarrow$ extract_query(y_i)
    docs $\leftarrow \mathcal{R}(q)$
  else if <answer> detected in y_i **then**
    stage $\leftarrow$ Terminal
    return extract_answer(y_i)
  else if <evidence> detected in y_i **then**
    stage $\leftarrow$ Reasoning
  end if
end for
return final response y

Table 2: Main Results (%) on Five benchmarks (the number of queries used for training is indicated in brackets). "*" indicates the statistically significance (i.e., two-sided t-test with $p < 0.05$) over the best baseline. Two most important columns: the averaged EM and F_1 are highlighted

Type	Method	PopQA		HotpotQA		2WikiMulti		Bamboogle		MuSiQue		Avg.	
		EM	F_1	EM	F_1	EM	F_1	EM	F_1	EM	F_1	EM	F_1
Zero-shot	Naïve Generation	12.7	16.5	15.7	24.8	20.2	28.0	6.4	17.4	2.7	10.2	11.5	19.4
	Standard RAG	38.4	44.7	29.3	39.9	29.4	36.3	17.6	24.1	6.7	15.1	24.3	32.0
Active	FLARE	14.3	17.6	18.1	25.7	27.9	32.8	12.0	20.8	4.3	12.6	15.3	21.9
	Self-RAG(146k)	22.7	33.9	21.0	29.7	12.0	25.2	1.6	10.9	4.6	13.3	12.4	22.6
Adaptive	AdaptiveRAG(3k)	36.6	41.5	29.1	40.7	24.2	33.4	18.4	26.1	6.9	14.3	23.0	31.2
RAG-CoT	Iter-Retgen	38.7	44.9	30.3	42.1	31.2	38.7	19.2	26.4	7.7	14.2	25.4	33.3
	IRCoT	36.2	43.6	27.7	41.5	23.5	32.5	17.2	22.5	8.6	13.2	22.6	30.7
Summary	RECOMP	40.5	45.8	29.7	41.2	33.2	39.4	21.7	28.6	9.2	15.8	26.9	34.2
	LongLLMLingua	39.2	45.1	31.4	43.2	34.5	40.2	20.3	27.4	8.7	14.9	26.8	34.2
	Selective-Context	34.9	41.5	19.3	27.3	20.3	29.7	15.3	22.6	6.1	13.7	19.2	27.0
Reasoning	Search-o1	33.2	40.3	24.8	38.1	16.4	27.1	30.4	40.6	6.3	13.7	22.2	31.96
	AutoRAG(10k)	38.6	44.1	33.3	43.7	39.5	46.1	24.8	32.2	11.3	18.3	29.5	36.9
	Search-R1(90k)	39.7	44.8	37.0	47.0	41.4	48.0	32.0	43.8	**14.6**	19.9	32.8	40.7
	ReasonRAG(5k)	**41.5***	**46.2***	**38.4***	**48.9***	**43.6***	**50.4***	**36.0***	**45.5***	12.8	**20.6***	**34.4***	**42.3***

In summary, ReasonRAG supports multi-step, flexible reasoning while maintaining structured decision control. The use of explicit placeholders enhances interpretability and facilitates programmatic control during deployment. The complete algorithmic flow is provided in Algorithm 1.

3 Experiments

3.1 Experimental Setup

Evaluation Dataset & Metrics. We evaluate ReasonRAG and all baselines on five public benchmarks: the single-hop QA dataset PopQA [34] and four multi-hop QA datasets, including HotpotQA [35], 2WikiMultiHopQA [36], Bamboogle [37], and MuSiQue [38]. Bamboogle and MuSiQue serve as out-of-domain QA evaluation datasets. The diversity of these datasets enables a comprehensive assessment of agentic RAG. We report Exact Match (EM) and F_1 scores as evaluation metrics. Refer to Appendix D for more details about dataset introduction, statistics, and metrics.

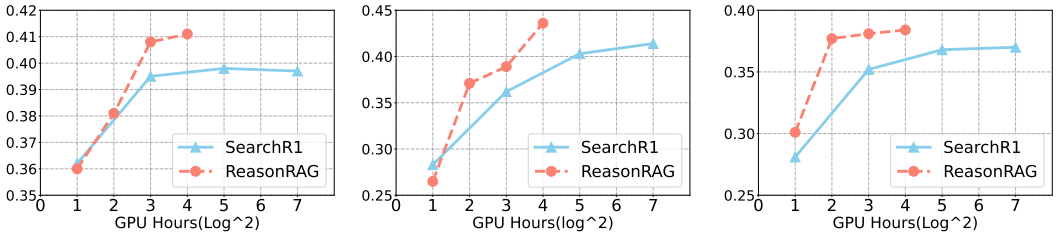
Baselines. We implement 12 baseline models which can be categorized into 6 types as follows: **Zero-shot:** Directly use prompt engineering on LLM to answer the question without or with retrieved documents [39]. **Active:** Actively make additional retrieval when retrieved data or generated responses have low confidence [40, 41]. **Adaptive:** Dynamically chooses the most suitable RAG pipeline from no-retrieval, single-hop, or multi-hop retrieval strategies [42]. **RAG-CoT:** Integrates chain-of-thought reasoning with retrieval, enabling multi-step, evidence-seeking answers [43, 44]. **Summary:** Compresses or summarizes retrieved content to fit model input constraints while retaining key information [45, 46, 47]. **Reasoning:** Enhances multi-hop reasoning by structuring the reasoning process and scrutinizing retrieved evidence [48, 27, 49]. Note that ReasonRAG and all baselines use Qwen2.5-7B-Instruct [50] as the backbone model, ensuring fair comparison. Refer to more implementation details about ReasonRAG and baselines in Appendix E.1 and E.2.

3.2 Main Results

We present detailed performance results on ReasonRAG against 12 baselines across five benchmark datasets, as shown in Table 2. Our key findings are summarized below:

- **Data Efficiency:** ReasonRAG, despite being trained on only 5k queries, outperforms the search-R1 baseline trained with 90k queries. On average across all datasets, ReasonRAG achieves higher EM

https://doi.org/10.52202/085713-1965



(a) PopQA (b) 2WikiMultiHopQA (c) HotpotQA

Figure 5: Training cost and convergence speed comparison (EM%) for ReasonRAG and Search-R1

Table 3: Impact of different optimization strategies on ReasonRAG 's effectiveness.

Method	PopQA		HotpotQA		2WikiMulti		Bamboogle		MuSiQue		Avg.	
	EM	F_1	EM	F_1	EM	F_1	EM	F_1	EM	F_1	EM	F_1
ReasonRAG (Base)	35.6	42.7	23.7	38.2	15.2	28.9	28.0	38.7	7.7	15.4	22.0	32.8
ReasonRAG (SFT)	31.6	37.4	26.8	38.7	35.1	40.9	17.6	27.3	8.6	15.5	23.9	32.0
ReasonRAG (RL-ORL): 5k queries	23.0	30.9	28.1	32.6	32.0	43.8	17.5	24.1	5.9	13.1	21.3	28.9
ReasonRAG (RL-ORL): 10k queries	39.5	45.7	36.7	46.7	40.5	47.2	30.7	40.6	12.6	19.5	32.0	39.9
ReasonRAG (RL-PRL)	**41.5**	**46.2**	**38.4**	**48.9**	**43.6**	**50.4**	**36.0**	**45.5**	**12.8**	**20.6**	**34.5**	**42.3**

(34.4%) and F_1 (42.3%) scores compared to search-R1 (32.8% EM, 40.7% F_1), highlighting the superior data efficiency of ReasonRAG. This demonstrates the effectiveness of process-supervised RL, which leverages fine-grained rewards, over current outcome-supervised methods.

- **Multi-hop Reasoning:** ReasonRAG shows substantial performance gains on multi-hop reasoning tasks. On the HotpotQA dataset, it achieves an F_1 score of 48.9%, outperforming models like AutoRAG (43.7% F_1) and search-R1 (47.0% F_1), both of which are trained on larger datasets. This underscores ReasonRAG 's strength in handling complex, multi-step questions that require integrating evidence from multiple sources.
- **Out-of-domain Generalization:** ReasonRAG demonstrates strong generalization to out-of-domain data. On challenging benchmarks such as Bamboogle and MuSiQue, it consistently achieves higher F_1 scores relative to other baselines. This indicates improved robustness and transferability of its reasoning capabilities across different domains.

3.3 Training Efficiency

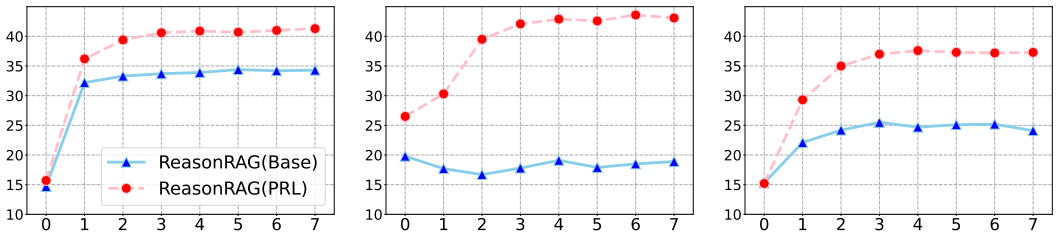
Figure 5 compares the training efficiency of ReasonRAG and Search-R1. The figure illustrates the progression of EM scores with increasing GPU hours across three datasets. The results reveal that ReasonRAG has higher training efficiency compared to Search-R1. ReasonRAG achieves superior EM scores with fewer GPU hours, indicating that it requires less training data and compute to reach strong performance levels. In contrast, Search-R1 requires significantly more GPU hours to reach similar performance.

The efficiency gap between the two models is particularly notable on multi-hop question answering tasks. For the single-hop PopQA dataset, performance gains for both models are comparably rapid as training progresses. However, for multi-hop datasets such as 2WikiMultiHopQA and HotpotQA, ReasonRAG consistently demonstrates significant improvements with increased GPU hours. This further underscores its effectiveness on complex reasoning tasks, where it delivers faster and greater performance improvements with fewer resources.

3.4 Effectiveness of Different Optimization Strategies

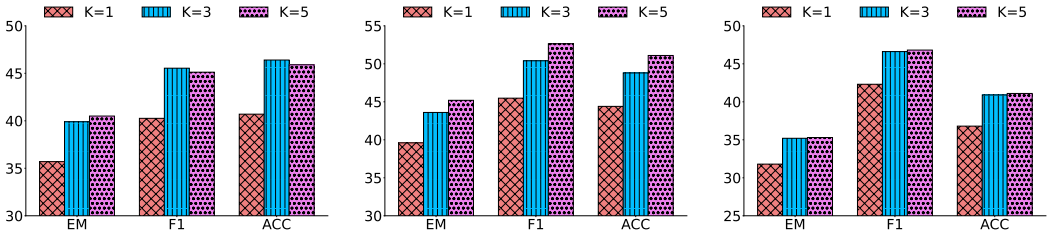
In this section, we compare the effectiveness of ReasonRAG utilizing three different optimization strategies against the base model. Our default approach, ReasonRAG (RL-PRL), is trained with process-level rewards as described in Section 2. For ReasonRAG (RL-ORL), we adopt outcome-level reward training following the Search-R1 protocol [27]. Specifically, we evaluate two versions:

https://doi.org/10.52202/085713-1965



(a) PopQA (b) 2WikiMultiHopQA (c) HotpotQA

Figure 6: The EM performance across varying retrieval iterations on 3 benchmarks.



(a) PopQA (b) 2WikiMultiHopQA (c) HotpotQA

Figure 7: Effect of top-k retrieved documents on ReasonRAG's performance across 3 datasets.

RL-ORL-5k, trained on the same 5k queries as RL-PRL, and RL-ORL-10k, which incorporates an additional 5k queries sampled from PopQA, HotpotQA, and 2WikiMultiHopQA, totaling 10k training examples. For ReasonRAG (SFT), we use the preferred responses from the RAG-ProGuide preference pairs as ground truth and apply supervised fine-tuning (SFT) via next-token prediction. Table 3 summarizes the performance of these four variants. Our main findings are as follows:

- **Superiority of PRL**: ReasonRAG (PRL) consistently outperforms all other variants across all datasets, both in-domain and out-of-domain, indicating stronger generalization capabilities.
- **High Data Demand of ORL**: ReasonRAG (ORL) achieves the second-best results, but requires substantially more training data to match the comparable performance of PRL. Although ORL is more effective than Base and SFT, its training efficiency is relatively low.
- **Overfitting in SFT**: SFT leads to overfitting on multi-hop reasoning paths, resulting in reduced performance on single-hop tasks. Furthermore, SFT-trained models generalize poorly, as demonstrated by a marked performance decline on the Bamboogle dataset.

3.5 Impact of Search on Performance

Performance vs. Retrieval Steps. Figure 6 shows the EM performance of ReasonRAG across varying retrieval iterations on 3 datasets. We observe a consistent trend: performance improves with more retrieval iterations and then gradually saturates. Notably, ReasonRAG can adaptively determine the required inference depth according to task complexity. For the single-hop PopQA dataset, performance converges within 2 to 3 retrieval steps, whereas more complex multi-hop tasks such as 2WikiMultiHopQA and HotpotQA require 3 to 5 steps to reach peak performance. In contrast, ReasonRAG (base) without preference optimization only achieves reliable gains on PopQA and struggles to handle multi-hop settings. These results demonstrate ReasonRAG 's ability to perform adaptive reasoning in response to the complexity of the input question.

Performance vs. Number of Retrieved Documents. Figure 7 compares the performance of ReasonRAG under different top-k retrieval settings, where k refers to retrieving top-k relevant document passages per search query. Results indicate that while ReasonRAG remains robust across a range of k values, its performance is sensitive to the quantity of retrieved information. With $k = 1$, limited context restricts the model's reasoning ability. Increasing k to 3 yields significant improvements across all datasets, suggesting that ReasonRAG effectively leverages additional evidence. Further increasing k to 5 does not further increase the performance on PopQA and HotpotQA, whereas

https://doi.org/10.52202/085713-1965

2WikiMultiHopQA continues to benefit from richer retrieved context. Overall, these findings highlight ReasonRAG 's capacity to utilize additional retrieved documents, particularly in more complex multi-hop scenarios.

4 Related Works

Prompt-Based Agentic RAG. Early prompt-based approaches leverage manually designed workflows to elicit the inherent capabilities of LLMs for interacting with external information. Specifically, the RAG task is often decomposed into subtasks such as adaptive retrieval judgment [42], query generation [21, 43, 44], evidence extraction [41, 45, 47, 46], and answer generation [40]. While some efforts have focused on optimizing RAG through personalization [51], graph-based retrieval [52], or reranking techniques [53], yet a critical gap remains in designing LLMs that can autonomously invoke search engines. Recently, agentic RAG aims to design workflows that empower LLM to autonomously interact with external information. OpenResearcher [54], AirRAG [55], IterDRAG [56], PlanRAG [57], and Search-o1 [49] demonstrate strong performance improvement by the effective incorporation with the search engine. Nevertheless, these methods are limited by their dependence on inherent capabilities for interacting with external information and the requirement for manual design when applied to new domains, and lack explicit mechanisms for eliminating distracting information [58].

RL-Based Agentic RAG. Reinforcement Learning (RL) has consistently delivered significant performance gains across a spectrum of sequential decision-making tasks, as evidenced by its successful application in domains such as recommender systems [59, 60, 61, 62, 63]. Recently, the success of models like DeepSeek-R1 [26] has vividly demonstrated the substantial potential of outcome-supervised RL in enhancing complex reasoning capabilities, establishing it as a mainstream paradigm for end-to-end optimization of LLMs. Following the widespread adoption of RL by major AI companies to improve the reasoning abilities of their models on complex tasks [64, 65, 66], recent work [27, 67, 68, 69] has extended outcome-supervised reinforcement learning to RAG, empowering LLMs to autonomously utilize search engines for intricate inference. While outcome-supervised RL has demonstrated significant performance gains, it also faces challenges such as reward sparsity, training instability, and substantial computational cost. Moreover, outcome-supervised RL typically demands extensive training resources. In contrast, process-supervised RL has recently been applied to enhance reasoning abilities, outperforming outcome-supervised approaches by providing fine-grained rewards [70, 71, 72, 73]. As an alternative avenue for improving LLM reasoning in RAG, process-supervised RL for RAG remains unexplored.

5 Conclusion

We introduce ReasonRAG, a process-supervised agentic RAG method for fine-grained policy optimization. Our approach integrates Monte Carlo Tree Search (MCTS) with the agentic RAG workflow to generate RAG-ProGuide, a high-quality dataset providing process-level supervision by prioritizing the shortest reasoning paths leading to correct answers. Leveraging RAG-ProGuide, we perform preference-based policy optimization to enhance LLMs' autonomous capabilities in query generation, evidence extraction, and answer synthesis. Experiments demonstrate that ReasonRAG achieves superior performance on five benchmark datasets using only 5k training instances, significantly fewer than the 90k required by Search-R1, highlighting the effectiveness of RAG-ProGuide 's high-quality process-level rewards in optimizing agentic RAG policies.

Acknowledgements

This research was partially supported by National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council's Research Impact Fund (No.R1015-23), Research Grants Council's Collaborative Research Fund (No.C1043-24GF), Research Grants Council's General Research Fund (No.11218325), Graduate Research Fund of the School of Economics and Management of Dalian University of Technology (No. DUTSEMDRFKO1), Institute of Digital Medicine of City University of Hong Kong (No.9229503), and Huawei (Huawei Innovation Research Program).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. Are large language models good statisticians? In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[5] Javier González and Aditya Nori. Does reasoning emerge? examining the probabilities of causation in large language models. *Advances in Neural Information Processing Systems*, 37:117737–117761, 2024.

[6] Wenlin Zhang, Chuhan Wu, Xiangyang Li, Yuhao Wang, Kuicai Dong, Yichao Wang, Xinyi Dai, Xiangyu Zhao, Huifeng Guo, and Ruiming Tang. Llmtreerec: Unleashing the power of large language models for cold-start recommendations. In *31st International Conference on Computational Linguistics (COLING 2025)*, pages 886–896. Association for Computational Linguistics, 2025.

[7] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

[8] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[9] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

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

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2, 2023.

[12] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 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*, pages 6491–6501, 2024.

[13] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251, 2024.

[14] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

[15] Google. Gemini deep research. 2024.

 https://doi.org/10.52202/085713-1965

[16] OpenAI. Deep research system card. *Technical Report*, 2025.

[17] Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Deep research: A survey of autonomous research agents. *arXiv preprint arXiv:2508.12752*, 2025.

[18] Maojia Song, Shang Hong Sim, Rishabh Bhardwaj, Hai Leong Chieu, Navonil Majumder, and Soujanya Poria. Measuring and enhancing trustworthiness of llms in rag through grounded attributions and learning to refuse. In *The Thirteenth International Conference on Learning Representations*.

[19] Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. In *The Twelfth International Conference on Learning Representations*.

[20] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations*.

[21] Reinald Kim Amplayo, Kellie Webster, Michael Collins, Dipanjan Das, and Shashi Narayan. Query refinement prompts for closed-book long-form qa. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7997–8012, 2023.

[22] Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5613–5626, 2024.

[23] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations*.

[24] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[25] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

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

[28] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[29] Yuqing Du, Alexander Havrilla, Sainbayar Sukhbaatar, Pieter Abbeel, and Roberta Raileanu. A study on improving reasoning in language models. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*, 2024.

[30] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

[31] Jeremy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Learning from natural language feedback. In *ACL Workshop on Learning with Natural Language Supervision*, 2022.

[32] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[33] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[34] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 7, 2022.

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

[36] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[37] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

[39] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

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

[41] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

[42] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong-Cheol Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 7036–7050. Association for Computational Linguistics, 2024.

[43] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[44] Zhihong Shao, Yeyun Gong, Minlie Huang, Nan Duan, Weizhu Chen, et al. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[45] Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

 https://doi.org/10.52202/085713-1965

[46] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, 2024.

[47] Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, 2023.

[48] Tian Yu, Shaolei Zhang, and Yang Feng. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*, 2024.

[49] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025.

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

[51] Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, et al. A survey of personalization: From rag to agent. *arXiv preprint arXiv:2504.10147*, 2025.

[52] Derong Xu, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Maolin Wang, Qidong Liu, Xiangyu Zhao, Yichao Wang, Huifeng Guo, Ruiming Tang, et al. Align-grag: Reasoning-guided dual alignment for graph retrieval-augmented generation. *arXiv preprint arXiv:2505.16237*, 2025.

[53] Pengyue Jia, Derong Xu, Xiaopeng Li, Zhaocheng Du, Xiangyang Li, Yichao Wang, Yuhao Wang, Qidong Liu, Maolin Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. Bridging relevance and reasoning: Rationale distillation in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4242–4256, 2025.

[54] Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, et al. Openresearcher: Unleashing ai for accelerated scientific research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 209–218, 2024.

[55] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Jingyi Song, and Hao Wang. Airrag: Activating intrinsic reasoning for retrieval augmented generation via tree-based search. *arXiv preprint arXiv:2501.10053*, 2025.

[56] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024.

[57] Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. Plan rag: Efficient test-time planning for retrieval augmented generation. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

[58] Yuanwei Liu, Nian Liu, Yi Wu, Hisham Cholakkal, Rao Muhammad Anwer, Xiwen Yao, and Junwei Han. Ntrenet++: Unleashing the power of non-target knowledge for few-shot semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2024.

[59] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pages 95–103, 2018.

[60] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for list-wise recommendations. In *1st Workshop on Deep Reinforcement Learning for Knowledge Discovery (DRL4KDD 2019)*, 2019.

[61] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1040–1048, 2018.

[62] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, and Hui Liu. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 750–758, 2021.

[63] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. Multi-task recommendations with reinforcement learning. In *Proceedings of the ACM web conference 2023*, pages 1273–1282, 2023.

[64] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[65] DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qinqing Zheng. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *The Thirteenth International Conference on Learning Representations*, 2024.

[66] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

[68] Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Otc: Optimal tool calls via reinforcement learning. *arXiv preprint arXiv:2504.14870*, 2025.

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

[70] Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S2r: Teaching llms to self-verify and self-correct via reinforcement learning. *arXiv preprint arXiv:2502.12853*, 2025.

[71] Zongyu Lin, Yao Tang, Xingcheng Yao, Da Yin, Ziniu Hu, Yizhou Sun, and Kai-Wei Chang. Qlass: Boosting language agent inference via q-guided stepwise search. *arXiv preprint arXiv:2502.02584*, 2025.

[72] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[73] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

[74] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[75] Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576, 2024.

 https://doi.org/10.52202/085713-1965

Checklist

In this section, we answer to the questions required by checklist stated in `https://neurips.cc/public/guides/PaperChecklist` as follows:

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims in the abstract and introduction match with the main contribution.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix H

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:[NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the experimental details in Appendix E

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

 https://doi.org/10.52202/085713-1965

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the dataset link and code link in Appendix E

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental settings and details in Section 3, more details are reported in Appendix E

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the statistical significance in table 2

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the experimental compute resources in Section 3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix I

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/085713-1965

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cite all code, data, and models in paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our dataset and introduce the details in Section 2.2.3

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

https://doi.org/10.52202/085713-1965

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: See Appendix F

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix Overview

The appendix includes the following sections:

- **Appendix A**: Details the MCTS tree construction process using process-level rewards. This serves as an extended explanation to Section 2.2.2.
- **Appendix B**: Illustrates the example of LLM evaluation for process-level simulation. This serves as an extended example of MCTS node expansion in Section 2.2.2.
- **Appendix C**: Compares the reasoning response between ReasonRAG(base) and ReasonRAG(PRL). This serves as an extended example for Section 3.4.
- **Appendix D**: Provides additional details of the evaluation setup. This serves as a supplement to Section 3.1.
- **Appendix E**: Provide more details of the implementation. This serves as a supplement to Section 3.1.
- **Appendix F**: Details the prompt design for Agentic RAG. This serves a supplement to Section 2.4.
- **Appendix G**: Details the licensing terms and conditions governing the use and distribution of the proposed datasets.
- **Appendix H**: Discusses the limitations and constraints of the proposed approach.
- **Appendix I**: Evaluates the potential societal implications and ethical considerations of the research.

A Monte Carlo Tree Search for Process-level Reward

Formally, for each RAG intermediate process, its corresponding state s encompasses the original question x, the preceding thoughts $y_{<i}$, and a stage $\in \{\text{Reasoning, Grounding, Terminal}\}$. The stage indicates the current decision mode within the reasoning process. In the Reasoning stage, the LLM autonomously decides whether to generate a new query or directly produce an answer. If a query is chosen, it triggers a call to the external search engine, and the retrieved documents are added to the context in the next step. If the model opts to generate an answer instead, the process transitions into the Terminal stage. In the Grounding stage, the model extracts relevant evidence spans from the retrieved documents based on the most recent query. After extracting evidence, the state transitions back to the Reasoning stage, enabling further iterative reasoning.

Based on the state, the policy of the next action is defined as

$$\pi(a|s) = \text{LLM}(a|s) = \begin{cases} \pi_\theta(\cdot|x, y_{<i}, p_{\text{stage}}), & \text{if stage is Reasoning} \\ \pi_\theta(\cdot|x, y_{<i}, \text{docs}, p_{\text{stage}}), & \text{otherwise} \end{cases} \tag{4}$$

Consequently, the state transition can be represented as $s_{t+1} = \text{concatenate}(s_t, a_t)$. Each node in the tree contains the following information: $\{N(s), Q(s), \text{Stage}(s)\}$, where $N(s)$ denotes the number of times state s has been visited, $Q(s)$ represents the current intermediate annotation collected through the Monte Carlo method, with values in the range $[0, 1]$. With the tree structure defined, the MCTS begins from the root node and constructs the tree by iteratively performing three key operations: selection, expansion, and backpropagation.

Selection: This step aims to select nodes that balance the search quality and exploration degree. The node selection starts from the root node, and iteratively selects the child nodes based on their state value Q and visiting frequency N. These variable are refined during the search strategy, detailed in the backpropagation section. To effectively trade off between exploring unvisited nodes and exploiting nodes with higher state value, we iteratively search for the next node using UCT score [74]. The state is selected according to the following formula:

$$s_i^* = \text{argmax}_{s_i \in \text{children(s)}}[Q(s_i) + c_{uct}\frac{\sqrt{\sum_i N(s_i)}}{1 + N(s_i)}] \tag{5}$$

where c_{uct} is a trade-off parameter to control the exploration degree. The algorithm begins by exploring unvisited states and progressively favors nodes with higher Q-values and fewer visits.

Expansion: Given a selected node that does not reach a terminal state and maximum child node limit, the expansion step proceeds with a single step of RAG reasoning based on Equation (4) and initializes a new child node with the generated response. Following response generation, the simulation step iteratively reasons until a final answer is reached, serving as the basis for initializing the reward of the created node. However, simulating RAG leads to inefficiency due to its need for iterative LLM reasoning and retrieval. To address this challenge, the correctness of the intermediate reasoning process is evaluated by LLM judgment based on the intermediate process against the golden answer, outputting a correctness value $v \in [0, 1]$, as defined in Equation (6). This approach avoids time-consuming simulations, providing an efficient evaluation for exploring the complex RAG reasoning space.

$$v(s) = \pi_\theta(\cdot|x, y_{1:t}, \text{answer}, p_{\text{eval}}) \tag{6}$$

Backpropagation: Following node expansion, the node statistics are iteratively updated from the current node back to the root node. The correctness score $Q(s_t)$ is updated based on Eq. (1). The visit count for the traversed nodes is incremented:

$$N(s) \leftarrow N(s) + 1 \quad \forall s \in \text{path from } s_t \text{ to root} \tag{7}$$

B Process Evaluation

The three cases illustrated in Figures 8–10 demonstrate the effectiveness of `ReasonRAG` in supporting stepwise reasoning over retrieved content. In the partially correct case, the agent retrieves relevant evidence but fails to cover all required entities, leading to incomplete inference. In the incorrect case, the agent misidentifies directors, resulting in a reasoning chain based on factual errors. In contrast, the fully correct case shows `ReasonRAG`'s ability to decompose the question into sequential sub-queries, accurately retrieve biographical information, and synthesize the correct answer through structured comparison. Although minor imperfections exist in the query formulation, the reasoning trace demonstrates clear progression, factual grounding, and answer justification. These cases collectively highlight `ReasonRAG`'s advantage in aligning sub-question formulation with retrieval intent and maintaining semantic coherence across reasoning steps, leading to improved robustness and interpretability in complex multi-hop scenarios.

C Case Study

Figure 11 shows an example for agentic RAG response from `ReasonRAG`. Before the policy optimization, LLMs fails to generate the appropriate query and mislead by irrelevant information. In contrast, the process-supervised RL empower LLMs to autonomously invoke query generation, evidence extraction, and answer generation.

https://doi.org/10.52202/085713-1965

Case of LLM judge partial correct process

Question
"Which film has the director died later, Il Coraggio or Shark Monroe?"
Golden Answers
"Il Coraggio."

Agent Thoughts:
So the next query is <query> What is the release year of Il Coraggio and who directed it? </query> Based on the query, the relevant evidence is <evidence>Il Coraggio is a 1955 Italian comedy film directed by Domenico Paolella. </evidence>

LLM judge

The agent's reasoning is severely incomplete and structurally flawed. It begins with a partial query that only addresses Il Coraggio but completely ignores Shark Monroe, (Omitted for brevity)

Moreover, the query does not directly request the death dates of the directors, which is the core requirement (Omitted for brevity)

While the evidence retrieved is correct and related to Il Coraggio, it is insufficient (Omitted for brevity)

So the score is 15.

Figure 8: An example of agent partial correctness in a RAG process.

Case of LLM judge incorrect process

Question
"Which film has the director died later, Il Coraggio or Shark Monroe?"
Golden Answers
"Il Coraggio."

Agent Thoughts:
So the next query is <query>When did the directors of Il Coraggio and Shark Monroe die?</query> Based on the query, the relevant evidence is: <evidence>Tonino Ricci, the director of Il Coraggio (also known as The Liberators), died on 9 March 2014. Umberto Lenzi, the director of Shark Monroe (also known as Man from the Deep River), died on 19 October 2017.</evidence>

LLM judge

The agent's reasoning shows a series of critical errors. First, the query itself is flawed: it misattributes directors to both films. Il Coraggio (1955) was directed by Domenico Paolella, not Tonino Ricci. (Omitted for brevity)

This fundamental misidentification invalidates the entire reasoning chain. The agent is not comparing the correct individuals, rendering the final answer baseless (Omitted for brevity)

Given these severe and foundational flaws in both retrieval and inference,(Omitted for brevity)

So the score is 5.

Figure 9: An example of agent incorrectness in a RAG process.

Case of LLM judge complete correct process

Question

‘‘Which film has the director died later, Il Coraggio or Shark Monroe?’’

Golden Answers

‘‘Il Coraggio.’’

Agent Thoughts:

So the next query is <query>What is the release year of Il Coraggio and who directed it?</query> Based on the query, the relevant evidence is <evidence>Il Coraggio is a 1955 Italian comedy film directed by Domenico Paolella./<evidence> So the next query is <query>Who directed Shark Monroe and when was the director’s death?</query> Based on the query, the relevant evidence is <evidence>William S. Hart directed Shark Monroe, which was released in 1918. William S. Hart died on December 3, 1945.</evidence>So the next query is <query>When did Domenico Paolella die?</query> Based on the query, the relevant evidence is <evidence>Domenico Paolella died on 7 October 2002. </evidence>So the answer is <answer>Il Coraggio</answer>

LLM judge

The agent’s reasoning process is largely correct in structure and intent: it decomposes the question into sub-queries that retrieve the directors of both films, their dates of death, and uses this information to compare who died later. This stepwise reasoning is sound and correctly arrives at the golden answer: Il Coraggio.
However, under strict scrutiny, several minor flaws emerge:

1. Imprecision in Initial Querying (Omitted for brevity)
2. Redundancy(Omitted for brevity)
3. Lack of Justification in the Final Step(Omitted for brevity)

Despite these imperfections, the agent correctly disambiguates the films and retrieves accurate information. (Omitted for brevity)

So the score is 88.

Figure 10: An example of agent complete correctness in a RAG process.

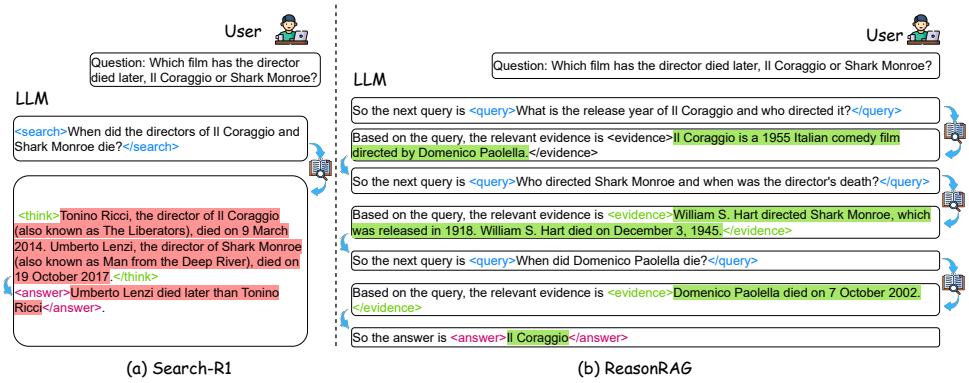


Figure 11: Case Study on 2WikiMultihopQA dataset.

https://doi.org/10.52202/085713-1965

D Evaluation Dataset Details

Evaluation Datasets. We use the process-level annotated questions (4,603 examples) as the training set for our policy optimization. These include 704 questions from PopQA, 2,843 from HotpotQA, and 1,056 from 2WikiMultiHopQA. For evaluation, we use the remaining unlabeled samples from PopQA as the test set, and we adopt the official development splits of HotpotQA and 2WikiMultiHopQA as test sets for multi-hop reasoning evaluation. Table 4 summarizes the training and test set sizes for each source. These datasets vary in their design focus and reasoning requirements. HotpotQA and 2WikiMultiHopQA are constructed to evaluate multi-hop reasoning capabilities, where answering a question requires combining information from multiple passages. HotpotQA includes sentence-level supporting facts and diverse question types, such as bridge and comparison questions. 2WikiMultiHopQA ensures genuine multi-step inference by leveraging structured knowledge from Wikidata and constructing explicit reasoning paths. PopQA, in contrast, is an open-domain QA dataset designed to probe factual recall in large language models. It focuses on a wide spectrum of factual knowledge, from high-frequency popular facts to long-tail, less commonly known information. By combining these datasets, we cover a diverse set of reasoning challenges, including factual retrieval, multi-hop inference, and process-level supervision.

Dataset Source	Train Set Size	Test Set Size
PopQA	704	11,267
HotpotQA	2,843	7,405
2WikiMultiHopQA	1,056	12,576
bamboogle	-	125
musique	-	2,417
Total	**4,603**	**33,790**

Table 4: Number of examples in the training and test sets for each dataset. Process-level annotations are used for training; test sets include remaining PopQA examples and official development splits of other datasets.

Evaluation Details. To evaluate model performance on question answering tasks, we adopt two standard metrics: Exact Match (EM) and F_1 score.

Exact Match (EM) measures the percentage of predictions that exactly match any of the reference answers. Formally,

$$\mathrm{EM} = \frac{1}{N}\sum_{i=1}^{N} \delta\left(y_i^{\text{pred}},\ y_i^{\text{gold}}\right), \tag{8}$$

where N is the number of examples, and $\delta(a, b) = 1$ if $a = b$, otherwise 0.

$\mathbf{F}_1$ score computes the token-level overlap between the predicted answer and the ground-truth answer. Let T_i^{pred} and T_i^{gold} denote the sets of tokens in the predicted and gold answers, respectively:

$$\mathrm{Precision}_i = \frac{|T_i^{\text{pred}} \cap T_i^{\text{gold}}|}{|T_i^{\text{pred}}|}, \tag{9}$$

$$\mathrm{Recall}_i = \frac{|T_i^{\text{pred}} \cap T_i^{\text{gold}}|}{|T_i^{\text{gold}}|}, \tag{10}$$

$$\mathrm{F}_1 = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i}. \tag{11}$$

We follow the official evaluation metrics implementation provided by the FlashRAG toolkit [75].

E Implementation Details

We summarize all artifacts (datasets, models, baselines, external knowledge base, etc) used in our experiments and their resource links in Table 5.

Name	Purpose	Artifact URL
PopQA	Eval Dataset	https://huggingface.co/datasets/RUC-NLPIR/FlashRAG_datasets/tree/main/popqa
HotpotQA	Eval Dataset	https://huggingface.co/datasets/RUC-NLPIR/FlashRAG_datasets/tree/main/hotpotqa
2WikiMultiHopQA	Eval Dataset	https://huggingface.co/datasets/RUC-NLPIR/FlashRAG_datasets/tree/main/2wikimultihopqa
Bamboogle	Eval Dataset	https://huggingface.co/datasets/RUC-NLPIR/FlashRAG_datasets/tree/main/bamboogle
MuSiQue	Eval Dataset	https://huggingface.co/datasets/RUC-NLPIR/FlashRAG_datasets/tree/main/musique
RAG-ProGuide	Train Dataset	https://anonymous.4open.science/r/ReasonRAG-B442.
BGE	Retriever	https://huggingface.co/BAAI/bge-base-en-v1.5
Wikidump 2018	Knowledge Source	https://archive.org/download/enwiki-20181220/enwiki-20181220-pages-articles.xml.bz2
Qwen2.5-7B-Instruct	Backbone Model	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
Adaptive-RAG	Baseline	https://huggingface.co/illuminoplanet/combined_flan_t5_xl_classifier
Self-RAG	Baseline	https://huggingface.co/selfrag/selfrag_llama2_13b
AutoRAG	Baseline	https://huggingface.co/ICTNLP/Auto-RAG-Llama-3-8B-Instruct
Search-R1	Baseline	https://huggingface.co/PeterJinGo/SearchR1-nq_hotpotqa_train-qwen2.5-7b-it-em-ppo
ReasonRAG	Our Method	https://anonymous.4open.science/r/ReasonRAG-B442.

Table 5: Resource links of artifacts used in our experiments.

E.1 Implementation Details of ReasonRAG

Following the setup in the FlashRAG toolkit, we use Wikidump 2018 as our knowledge source. To ensure retrieval quality, we augment our corpus by incorporating relevant content from the PopQA, HotpotQA, and 2WikiMultiHopQA datasets. All datasets are available on Huggingface. Subsequently, we employ BGE as our retriever, consistently retrieving the top 3 documents. For all methods not requiring fine-tuning, we use Qwen2.5-7B-Instruct as our baseline model for inference.

E.2 Implementation Details of Baselines

For baseline implementations, we utilize the FlashRAG [75] reproduction, where several models, such as Naïve Generation, Standard RAG, FLARE, Iter-Retgen, RECOMP, LongLLMLingua, and Selective-Context, require no additional parameter configuration. For Self-RAG, we use the checkpoint provided in the FlashRAG reproduction. For AdaptiveRAG, we employ the FlashRAG reproduction's router and qwen2.5-7b-instruct as the reasoning model. For AutoRAG, we conduct inference using the publicly available checkpoint from Hugging Face. For Search-R1, we use the reproduced qwen2.5-7b-base and qwen2.5-7b-instruct checkpoints for inference.

F Prompt Instructions

Our prompts include Reasoning, Grounding in Agentic RAG workflow, and a process evaluation prompt for node expansion. No extra prompt design is needed when input a new question into LLMs for inference. The prompt details are illustrated in Figure 12, Figure 13, and Figure 14.

G License Agreement

The RAG-ProGuide is constructed based on popqa, hotpotqa, and 2wikimultihopqa from FlashRAG dataset [75]. All these datasets are using CC-BY-SA-4.0 license, allowing the modification for research use. For the new constructed dataset RAG-ProGuide, including but not limited to the questions and process-level reward, we make them available solely for research purposes. Users are permitted to use, modify, and share these annotations for academic and non-commercial research activities. Any other use, including commercial exploitation, is not permitted without explicit written permission from the authors.

https://doi.org/10.52202/085713-1965

Reasoning

You are a question-answering assistant with access to a retrieval tool. Your goal is to provide a concise and accurate reasoning process.
Instructions:
* Error Reflection: If errors exist in previous thoughts, identify and correct them. Skip this step if no errors are present.
* Information Sufficiency: Evaluate whether the current information is sufficient to fully and accurately answer the question. If additional retrieval is needed, deconstruct the question and generate the next query. Avoid repeating previous queries. If no meaningful new query can be generated, explain why and provide an answer based on the current information. * Conciseness: Ensure both queries and answers are concise, using nouns or short phrases whenever possible.
* Conclusion:
If generating an answer: "So the answer is <answer>{answer_format}</answer>". If more retrieval is needed: "So the next query is <query>query</query>".

Figure 12: System prompt for Reasoning

Grounding

You are an information retrieval assistant. Your task is to extract relevant evidence from the provided Wikipedia documents based on the latest query.
Instructions:
* Identify key terms or concepts in the query. * Search the documents for evidence that supports the query. * Response format: If relevant evidence is found, output: Based on the query, the relevant evidence is <evidence>evidence</evidence>. If no relevant evidence is found, output: <evidence>None</evidence>.

Figure 13: System prompt for Evidence Extraction

Process Evaluation

An agent is tasked with answering a question using a retrieval tool. Critically assess its intermediate reasoning process to determine if it leads to the correct answer. Identify all flaws, inconsistencies, and mistakes in the thought process. Every imperfection, no matter how small, must be acknowledged. Evaluate how effectively the reasoning supports the final answer and the overall accuracy of the response. Ensure the evaluation is extremely harsh, leaving no leniency. Even if the answer seems close to correct, do not award full marks to maintain strict grading standards. Assign a score between [0, 1] based on the severity of flaws and the reasoning's accuracy in leading to the golden answer. Respond briefly and conclude with: So the score is [Score].

Figure 14: System prompt for Process Evaluation

H Limitations

We employ process-supervised RL to optimize the model policy. In contrast to outcome-supervised RL, our approach necessitates exploring process-level actions for fine-grained reward annotation. Consequently, acquiring sufficient data for process-level annotation incurs a higher time cost compared to outcome supervision during data rollout. Nevertheless, as the training efficiency verifies, our data exhibits superior quality, enabling models to achieve greater performance gains with fewer data samples.

I Societal Impacts

LLMs carry the risk of generating uncontrollable responses. When an LLM retrieves racist or harmful information from a search engine, it could be inadvertently led to produce similar content. We strongly advise users to employ our agentic RAG framework responsibly by integrating secure search engines or knowledge corpora and conducting evaluations within open-source, safe environments and datasets.

https://doi.org/10.52202/085713-1965