

Reflective RAG: Self-Evaluation Driven Strategy Optimization in Agentic Retrieval-Augmented Generation

Haiyan Wu, Chenchen Wang, Chaoqun Sun, Chengxiong Lu,
Yanhong Chen, Zhiqiang Zhang*, Xiaoqing Feng

School of Information Technology and Artificial Intelligence,
Zhejiang University of Finance and Economics, China

{wuhy2020,wccc,sunchaoqun,luchengxiong,cyhong,zqzhang,fenglinda}@zufe.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a widely adopted paradigm for grounding Large Language Models (LLMs) in external knowledge. Recent agentic RAG systems introduce multi-turn reasoning, but they often lack the capacity to evaluate the utility of retrieved information, leading to brittle reasoning and suboptimal decision-making. We propose Reflective RAG, an agentic framework that incorporates self-evaluation to dynamically optimize retrieval and generation strategy. At its core, Reflective RAG employs a reflection tagging mechanism that allows the model to critique the relevance of retrieved content, thereby explicitly guiding its subsequent policy. To ensure robust learning, we introduce a two-stage training procedure that partially decouples evaluation semantics from strategy optimization. First, during supervised fine-tuning (SFT), the model learns to generate accurate reflection signals by self-correcting labels based on internal uncertainty. Second, a reinforcement learning (RL) stage optimizes the agent’s strategy using these reflections, stabilized by dynamic KL regularization. Evaluations across five knowledge-intensive QA benchmarks demonstrate that Reflective RAG consistently outperforms strong agentic baselines. Further analysis demonstrates its improved training stability and stronger generalization to complex multi-hop reasoning tasks.

1 Introduction

Large language models (LLMs) have demonstrated exceptional proficiency in language understanding and generation across various natural language processing tasks (Zhao et al., 2023; Naveed et al., 2025). However, their reliance on static parametric knowledge limits their performance in knowledge-intensive domains, leading to information gaps or hallucinations when dealing with rare facts (Kandpal et al., 2023; Yu et al., 2023; Zhang et al., 2023).

*Corresponding author.

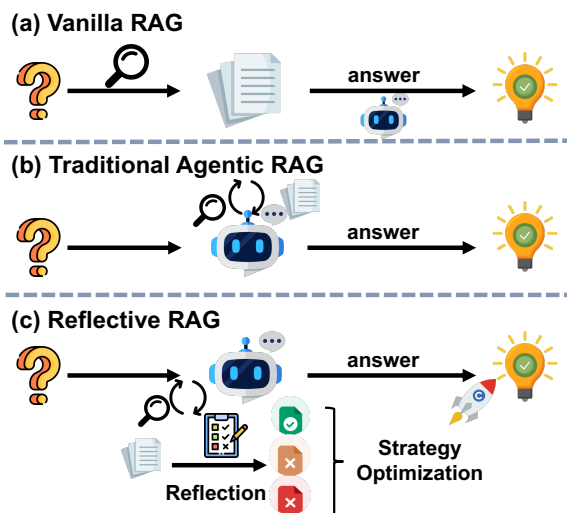


Figure 1: Comparison of retrieval and reflection strategies in RAG systems.

Retrieval-augmented generation (RAG) addresses these limitations by integrating external information with parametric knowledge, enhancing factual grounding and enabling more comprehensive, timely, and reliable outputs (Lewis et al., 2020; Peng et al., 2023; Fan et al., 2024).

However, vanilla RAG methods typically rely on a static “retrieve-then-generate” paradigm, performing retrieval based solely on the initial input. This lack of adaptive retrieval during generation severely limits their effectiveness in complex, multi-hop reasoning scenarios. Without the flexibility to dynamically determine when and what to retrieve, these methods struggle with information integration in multi-hop tasks. Recent studies have introduced interactive strategies within the reasoning process, including query decomposition (Press et al., 2023; Xu et al., 2024), self-correction mechanisms (Asai et al., 2024; Yan et al., 2024), and active retrieval (Jiang et al., 2023; Li et al., 2025). Furthermore, with the rise of reinforcement learning in LLM training (Zhang et al., 2025b), agen-

tic RAG has demonstrated improved reasoning by utilizing rule-based rewards derived from final outputs (Song et al., 2025; Jin et al., 2025). Although existing methods introduce reflection signals or rely on outcome-based optimization, they typically do not explicitly model evaluation as a structured and trainable component that directly constrains policy optimization during multi-turn reasoning, as illustrated in Figure 1. Therefore, assessment is often delegated to external expert models or decoupled from policy learning, limiting the model’s ability to identify redundant or misleading information and adapt its strategies in multi-turn reasoning. This disconnect can lead to premature commitment to final answers under high uncertainty, undermining the reliability of multi-turn retrieval reasoning.

To address these limitations, we present **Reflective RAG**, a general framework designed to assess retrieved content utility through reflection tagging and refine multi-turn retrieval and generation via explicit evaluation-guided strategy constraints. Reflective RAG is fully self-contained, alleviating dependence on external expert models. We introduce a bifurcated training process to ensure robust policy learning: (1) the supervised fine-tuning (SFT) stage, with the model self-correcting evaluation labels via internal likelihood signals and learning basic post-evaluation strategies; and (2) the reinforcement learning (RL) stage, applying evaluation-guided strategy constraints to regulate rollout trajectories and dynamic KL regularization to stabilize both evaluation semantics and policy updates. This decoupling ensures that the model develops interpretable assessment signals that offer direct guidance for reasoning. The contributions are summarized as follows:

- We propose Reflective RAG, an agentic RAG framework that incorporates explicit self-evaluation into multi-turn retrieval reasoning, and constructs high-quality training trajectories via self-corrected evaluation signals without external supervision (Section 3.2).
- We introduce an evaluation-aware optimization strategy that combines evaluation-guided strategy constraints with decaying KL regularization on evaluation labels (Section 3.3), enabling stable semantics and effective policy optimization in multi-turn reasoning.
- Extensive experiments (Table 1) on five knowledge-intensive QA benchmarks show

that Reflective RAG consistently outperforms strong baselines, and ablations as well as quantitative analyses further confirm its advantages in training stability and generalization.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-augmented generation enhances LLMs by incorporating external knowledge to improve factual grounding, completeness, and reliability (Lewis et al., 2020; Arslan et al., 2024). Prior work in RAG has primarily pursued two complementary directions: enhancing the effectiveness of retrievers and improving the capacity of LLMs to utilize retrieved knowledge effectively. Retriever-oriented approaches aim to improve retrieval relevance and alignment with LLM preferences (Glass et al., 2022; Shi et al., 2024; Jiang et al., 2025b), while LLM-centric methods explore task-adaptive fine-tuning (Zhang et al., 2024; Wei et al., 2024), self-reflection (Asai et al., 2024), and preference modeling to handle noisy or misleading information (Xu et al., 2025; Li et al., 2024). These methods enhance RAG by aligning retrieval with generation and filtering irrelevant content. However, most do not systematically model retrieval interactions in multi-hop reasoning and rely on expert-labeled or manually annotated trajectories. In contrast, Reflective RAG enables the model to self-correct evaluation labels and enforce strategy constraints, thereby adaptively optimizing reasoning strategies without external expert supervision.

2.2 Agentic RAG with Reinforcement Learning

Recently, with advances in agentic RAG (Singh et al., 2025), approaches have increasingly integrated retrieval with multi-hop reasoning, transforming retrieval from a static process into a dynamic one that can adaptively decide when and what to retrieve during reasoning. For example, ReAct (Yao et al., 2022) introduces a collaborative paradigm. SearChain (Xu et al., 2024) conducts multi-round retrieval by progressively decomposing questions, and Search-ol (Li et al., 2025) incorporates retrieval and information integration directly into the reasoning process. To further enhance models’ ability to interact with retrieval tools, inspired by DeepSeek-R1 (Guo et al., 2025), an increasing number of agentic RAG methods adopt reinforcement learning for training (Liang

et al., 2025; Zhang et al., 2025a). For instance, ReSearch (Chen et al., 2025), R1-Searcher (Song et al., 2025) and Search-R1 (Jin et al., 2025) rely solely on trajectory format and final answer correctness as reward signals. CARE (Wang et al., 2025) combines retrieval-aware rewards with curriculum learning. MT-PPO (Wei et al., 2025) extends trajectory-level rewards to turn-level optimization objectives. StepSearch (Zheng et al., 2025) proposes a token-level stepwise reward mechanism based on information gain and redundancy penalties. However, these methods generally lack explicit and trainable evaluation mechanisms of intermediate retrieval results and corresponding strategy constraints. In contrast, Reflective RAG explicitly models self-evaluation across multi-turn retrieval interactions and incorporates these evaluations as strategy constraints during rollout sampling, enabling stable optimization of evaluation-aware policies for multi-turn retrieval reasoning.

3 Methodology

This section introduces our agentic framework, Reflective RAG, as depicted in Figure 2. We first formalize the task in Section 3.1. Section 3.2 then describes self-correcting evaluation and strategy learning through supervised fine-tuning (SFT), followed by Section 3.3, which presents reinforcement learning (RL) with evaluation-guided strategy constraints and dynamic KL regularization over evaluation labels for stable training.

3.1 Task Definition

We consider multi-turn retrieval reasoning as an LLM agent interacting with a retrieval environment R and an external knowledge base D . At turn t , the agent maintains a trajectory of previous actions and observations $\tau_t = (a_1, o_1, \dots, a_t, o_t)$ and samples the next action according to a policy:

$$a_t \sim \pi_\theta(a_t \mid \tau_{t-1}). \quad (1)$$

At each time step, the action a_t constitutes either a retrieval action or the final answer generation. If a_t is a retrieval action, the agent interacts with the knowledge base D via the retriever R , yielding an observation $o_t = R(q_t, D)$, where the query q_t is derived from a_t . The interaction trajectory is subsequently updated as $\tau_t = (\tau_{t-1}, a_t, o_t)$. The process terminates at turn T , where the final action a_T produces the answer A .

Algorithm 1 Evaluation-aware Trajectory Construction for SFT

Require: Training questions \mathcal{Q} , base LLM π , retriever R , percentile λ
Ensure: SFT trajectory dataset \mathcal{D}_{SFT}

- 1: Initialize buffer $\mathcal{D} \leftarrow \emptyset$
- 2: **for** each $Q \in \mathcal{Q}$ **do**
- 3: Using π and R , sample trajectory $\tau_T = \{(a_1, o_1), \dots, (a_{T-1}, o_{T-1}), a_T\}$
- 4: **if** τ_T is well-formatted and answers correctly **then**
- 5: Add τ_T to \mathcal{D}
- 6: **end if**
- 7: **end for**
- 8: Initialize $\mathcal{D}_{\text{SFT}} \leftarrow \emptyset$
- 9: **for** each trajectory $\tau_T \in \mathcal{D}$ **do**
- 10: **for** each retrieval step t in τ_T **do**
- 11: Compute ΔPPL_t using τ_t and $\tau_t^{\setminus o_t}$
- 12: **end for**
- 13: Assign evaluation labels via λ -percentile threshold over $\{\Delta\text{PPL}_t\}$
- 14: **while** $\neg\text{ANSWER}(\tau_T)$ and $|\tau_T| < T_{\max}$ **do**
- 15: $e_t \leftarrow \text{CORRECTEVAL}(e_t \mid \tau_t)$
- 16: **if** $e_t \neq e_t^{\text{prev}}$ **then**
- 17: $\tau_{\geq t} \sim \pi(\cdot \mid \tau_t, e_t)$
- 18: **end if**
- 19: **end while**
- 20: **if** τ_T is well-formatted and answers correctly **then**
- 21: $\mathcal{D}_{\text{SFT}} \leftarrow \mathcal{D}_{\text{SFT}} \cup \{\hat{\tau}_T\}$
- 22: **end if**
- 23: **end for**

return \mathcal{D}_{SFT}

3.2 Supervised Fine-Tuning Phase

The supervised fine-tuning (SFT) stage equips the model with the ability to generate structured reasoning formats and to perform self-evaluation over multi-turn retrieval outcomes, while learning basic reasoning and retrieval strategies. As shown in Algorithm 1, this stage consists of three steps: trajectory sampling and filtering, self-correction of evaluation labels, and evaluation-aware trajectory reconstruction.

Trajectory Sampling and Filtering. We use the base model to sample complete multi-turn retrieval reasoning trajectories from the training data, forming an initial trajectory set. Each trajectory follows a unified structured format, where each step includes `<think>`, `<search>`, `<information>`, and `<evaluation>`, representing the current reasoning state, generated sub-query, retrieved external knowledge, and an assessment of its usefulness, respectively. When sufficient information is obtained, the model directly outputs `<answer>` after `<think>`.

Only trajectories with complete structured formats and correct final answers are retained for subsequent evaluation label correction and strategy learning. This filtering process reduces noise from

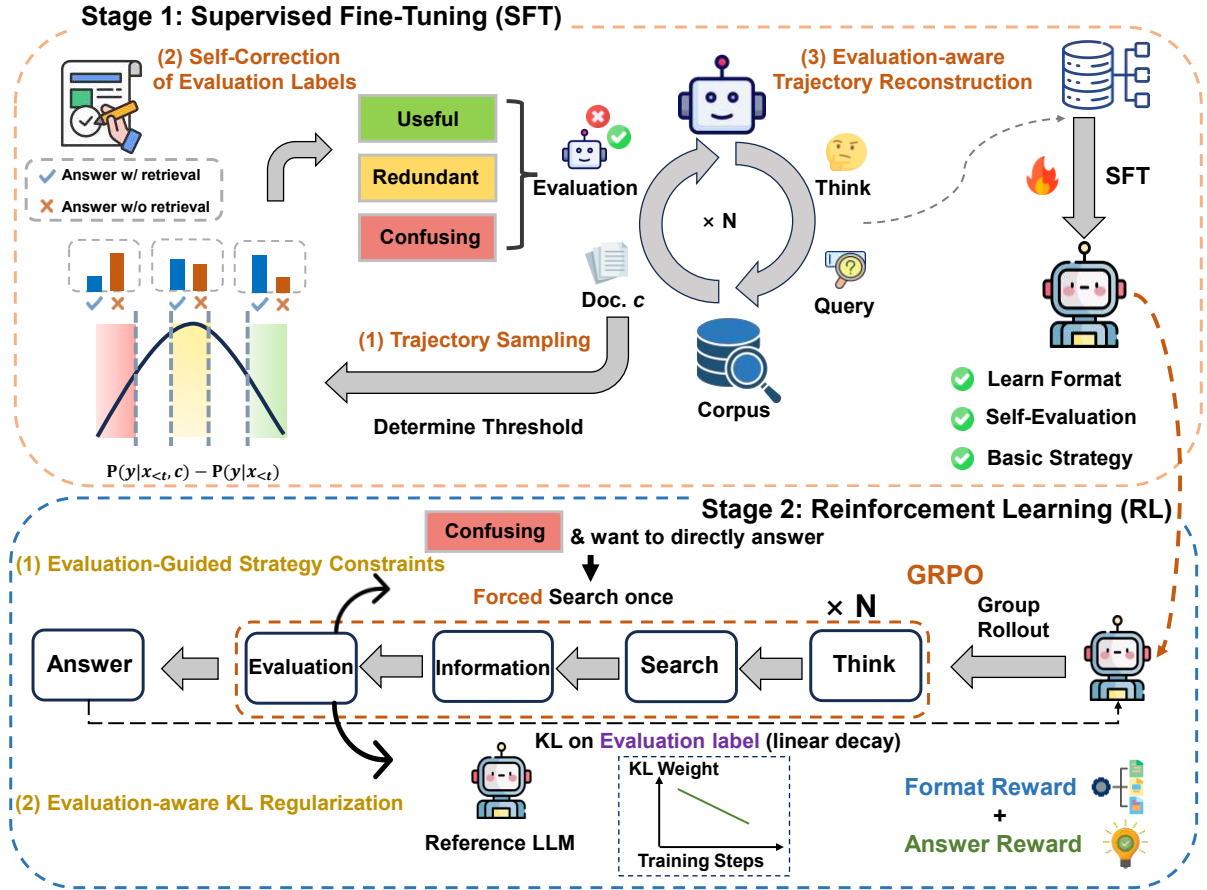


Figure 2: Overview of the Reflective RAG framework. **Stage 1:** Supervised Fine-Tuning (SFT) performs evaluation-aware trajectory construction via self-corrected labels. **Stage 2:** Reinforcement Learning (RL) optimizes the retrieval and generation policy with evaluation-guided strategy constraints and decayed KL regularization over evaluation tokens.

erroneous reasoning or formatting noise, producing high-quality, structurally consistent data for the following training phases.

Self-Correction of Evaluation Labels. To ensure consistency between evaluation labels and the model’s internal uncertainty signals at both semantic and decision-making levels, we perform self-correction of the generated labels based on changes in perplexity (PPL). Specifically, for the retrieved information o_t at step t , we compute the model’s perplexity in generating the gold answer A :

$$\text{PPL}(A | \tau_t) = \exp \left(-\frac{1}{|A|} \sum_{i=1}^{|A|} \log p(a_i | \tau_t) \right), \quad (2)$$

where τ_t denotes the complete interaction trajectory including the retrieved information at step t . The contribution of retrieval o_t is measured by the change in perplexity when it is removed:

$$\Delta\text{PPL}_t = \log \text{PPL}(A | \tau_t \setminus o_t) - \log \text{PPL}(A | \tau_t), \quad (3)$$

where $\tau_t \setminus o_t$ is the trajectory with the t -th retrieval information o_t removed. ΔPPL_t is computed by measuring the marginal contribution of o_t to the prediction of the final answer, conditioned on the preceding trajectory. A positive ΔPPL_t indicates that the retrieval reduces perplexity and benefits reasoning, whereas a negative value indicates potential interference (Jiang et al., 2025b).

After computing ΔPPL_t for all retrieval steps, we correct the evaluation labels based on their quantile distribution, which is applied only during SFT trajectory construction and is not used in actual reasoning at inference time. The detailed percentile-based criteria are provided in Appendix A.

Evaluation-aware Trajectory Reconstruction. After self-correcting the evaluation labels, we perform constrained resampling of subsequent reasoning steps to construct high-quality trajectories that align evaluation with model strategy. Specifically, for each retrieval step, the model adapts its strategy based on the corrected la-

bel: Useful—fully leverage the retrieved information; Redundant—attenuate reliance on it; Confusing—ignore it and trigger new reasoning or retrieval. The exact prompt templates used to implement these strategies are provided in Appendix D.2.

Leveraging corrected evaluation labels, we iteratively resample retrieval and reasoning steps to generate trajectories that align with these labels. This refined dataset is then employed for Supervised Fine-Tuning (SFT), effectively bridging the gap between evaluation metrics and the original generation strategy.

3.3 Reinforcement Learning Phase

After acquiring initial self-evaluation capabilities and a basic evaluation-guided strategy through SFT, we further optimize the model using reinforcement learning to enhance stability and policy consistency in multi-turn retrieval reasoning. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which maximizes trajectory-level rewards while constraining semantic drift in evaluation labels and stabilizing the model’s post-evaluation policy during rollout through evaluation-guided strategy constraints. The detailed GRPO objective and optimization procedure with our modifications are provided in Appendix B.

Evaluation-Guided Strategy Constraints. During rollout, we apply selective, evaluation-guided strategy constraints. Specifically, if a retrieval is labeled “Confusing” but the model attempts to directly generate the final answer, we enforce at least one additional retrieval or reasoning step to prevent premature termination under potentially misleading information. This constraint is also applied at inference to ensure more reliable reasoning. In contrast, for “Useful” and “Redundant” labels, no hard constraints are imposed, allowing the model to continue retrieval or generate an answer freely. This design minimizes intervention while preserving exploration flexibility.

Evaluation-aware KL Regularization. During GRPO, the model policy is continuously updated. Without proper constraints, the generation distribution of evaluation labels will drift from the semantics learned during SFT. To address this issue, we introduce a group-weighted KL regularization:

$$D_{\text{KL}} = \alpha(t) D_{\text{KL}}^{\text{eval}} + \beta D_{\text{KL}}^{\text{other}}, \quad (4)$$

where $D_{\text{KL}}^{\text{eval}} = D_{\text{KL}}(\pi_{\theta}^{\text{eval}} \parallel \pi_{\theta_{\text{ref}}}^{\text{eval}})$ and $D_{\text{KL}}^{\text{other}} = D_{\text{KL}}(\pi_{\theta}^{\text{other}} \parallel \pi_{\theta_{\text{ref}}}^{\text{other}})$ constrain the deviation of the

current policy π_{θ} from the reference policy $\pi_{\theta_{\text{ref}}}$. A larger weight $\alpha(t)$ is applied to tokens within the <evaluation> segment to preserve semantic stability, while a smaller weight β is assigned to all other tokens to maintain the overall language distribution.

The KL weight for evaluation tokens follows a linear decay schedule:

$$\alpha(t) = \beta + (\alpha_0 - \beta) \left(1 - \frac{t}{T_{\text{train}}}\right), \quad (5)$$

where α_0 denotes the initial weight and T_{train} is the total number of training steps. This design enforces stronger constraints in the early stage of RL to prevent evaluation drift, and gradually relaxes them in the later stage, allowing the model to adaptively refine its assessment of retrieval usefulness while maintaining sufficient policy flexibility.

Reward Design. Unlike most RL-based agentic RAG methods that rely on complex or externally engineered rewards, we adopt a sparse yet stable trajectory-level reward that focuses solely on whether the model produces a complete reasoning trajectory with a correct final answer and a valid output format:

$$R_{\text{total}} = \lambda_{\text{acc}} R_{\text{acc}} + \lambda_{\text{fmt}} R_{\text{fmt}}, \quad (6)$$

where R_{acc} measures answer correctness and R_{fmt} evaluates compliance with the required trajectory format. This design deliberately avoids rewarding evaluation labels directly, preventing the model from exploiting or manipulating them to achieve higher rewards (i.e., reward hacking), which could otherwise induce semantic drift in the evaluation signals.

4 Experiments

4.1 Datasets and Settings

Datasets. To further validate the performance of Reflective RAG, we evaluate it on five question answering (QA) datasets, including two open-domain benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019) and WebQuestions (Berant et al., 2013), as well as three multi-hop benchmarks: 2WikiMultiHopQA (2WikiMHQ) (Ho et al., 2020), HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022). Following Search-R1 (Jin et al., 2025), we adopt strict Exact Match (EM) as the primary evaluation metric and additionally report the F1 score to

Method	NQ		WebQ		2WikiMHQ		HotpotQA		Musique	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Models (3B Parameters)										
LLM w/o RAG	10.69	18.31	12.99	28.68	23.08	26.43	15.31	21.63	1.94	6.83
CoT	14.90	22.51	14.81	29.78	20.11	24.21	16.61	23.33	4.43	11.11
Vanilla RAG	25.90	35.21	15.65	30.25	22.20	26.52	20.69	27.97	3.48	7.93
Search-o1	24.27	33.53	16.29	32.14	21.84	27.24	21.05	29.29	5.79	12.95
Search-R1	33.41	41.38	26.13	39.26	27.55	32.01	25.63	33.85	5.17	11.51
Reflective RAG	35.96	44.05	29.04	42.80	27.58	32.57	28.44	37.76	7.07	14.09
Models (7B Parameters)										
LLM w/o RAG	16.98	24.93	20.28	34.84	24.75	29.01	19.19	25.75	3.89	9.30
CoT	18.70	28.81	17.18	33.80	25.56	30.24	20.89	29.27	5.50	13.77
Vanilla RAG	28.78	38.13	18.36	33.12	16.71	22.18	22.94	30.74	4.26	9.32
Self-RAG	31.36	39.05	17.22	31.90	12.77	21.78	14.14	24.37	3.64	9.91
Search-o1	29.70	39.96	18.45	35.08	20.67	26.61	25.54	35.26	7.82	15.29
s3	31.02	41.56	18.21	34.25	21.56	27.16	27.21	36.07	6.04	12.19
Search-R1	36.70	45.28	29.68	42.93	29.06	33.83	29.09	38.32	9.14	17.01
Reflective RAG	37.70	46.46	30.17	44.15	30.27	35.34	30.75	40.42	9.89	18.33

Table 1: Performance comparison of Reflective RAG and various baselines across five QA datasets (EM/F1), with the best results highlighted in **bold**.

provide a more fine-grained assessment of model robustness and effectiveness.

Settings. In the RAG system, we use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Qwen et al., 2025) as generator models. All experiments use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the external knowledge corpus, with e5-base-v2 (Wang et al., 2022) serving as the retriever. For fairness, we retrieve a single passage per query during trajectory generation. In the SFT stage, we set the evaluation label percentile threshold to $\lambda_e = 50$. In the RL stage, we set the initial KL weight for evaluation labels to $\alpha_0 = 0.005$, the KL weight for other tokens to $\beta = 0.001$, and the reward weights to $\lambda_{acc} = 0.8$ and $\lambda_{fmt} = 0.2$. Each model is trained once on the same training data and evaluated on all five datasets to assess generalization performance. Additional implementation details and prompt templates are provided in Appendix D.

4.2 Baselines

We evaluate Reflective RAG against a comprehensive suite of baselines representing different paradigms of reasoning and retrieval: 1) **CoT** (Wei et al., 2022): A reasoning-only baseline that performs multi-hop inference without external knowledge retrieval. 2) **Vanilla RAG** (Ram et al., 2023): The standard “retrieve-then-read” paradigm, where the generator produces an answer based on a sin-

gle initial retrieval step. 3) **Self-RAG** (Asai et al., 2024): An adaptive framework that utilizes reflection tokens for retrieval and refinement via Supervised Fine-Tuning (SFT), but notably lacks a reinforcement learning (RL) component for policy adaptation. 4) **Search-o1** (Li et al., 2025): A multi-turn retrieval baseline designed for information integration that does not incorporate RL optimization. 5) **s3** (Jiang et al., 2025a): A lightweight RL framework that decouples search from generation and learns data-efficient retrieval policies via a Gain-Beyond-RAG reward. 6) **Search-R1** (Jin et al., 2025): A recent agentic RAG approach that employs multi-turn retrieval and is trained via RL, using final answer correctness as the reward signal.

Furthermore, to ensure a fair comparison, we evaluate all baselines with Qwen2.5 as the backbone, except for Self-RAG, where we use the LLaMA2-7B checkpoint released by the authors.

4.3 Main Results

The results are reported in Table 1, from which we draw the following observations:

• **Superior Overall Performance.** Reflective RAG consistently outperforms all baselines across the five datasets. On the 3B scale, Reflective RAG achieves average improvements of 2.04% in EM and 2.65% in F1 over Search-R1. For 7B models, it yields gains of 1.02% in EM and 1.47% in F1.

Setting	NQ		WebQ		2WikiMHQ		HotpotQA		Musique	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Original	17.31	25.00	11.66	24.18	17.02	22.87	17.11	24.30	4.92	10.31
SFT Only	22.30	30.13	15.06	26.88	20.13	24.30	20.30	27.42	4.92	9.12
No correction	31.69	40.01	25.15	39.23	25.79	31.03	25.46	34.25	6.08	12.91
No Cons.	30.80	39.54	23.38	38.17	26.57	31.46	26.23	35.53	7.24	13.91
No eval. KL	34.49	42.57	27.61	41.16	28.97	33.63	27.33	36.58	6.62	13.51
Reflective RAG	35.96	44.05	29.04	42.80	27.58	32.57	28.44	37.76	7.07	14.09

Table 2: Ablation study of Reflective RAG components on five QA datasets using Qwen2.5-3B-Instruct. “Original” denotes the base model. “Cons.” indicates the evaluation-guided strategy constraints, and “eval. KL” refers to the KL regularization applied to evaluation labels as defined in Eq. 4.

Crucially, Reflective RAG achieves these improvements without external expert models or additional RL sampling. We attribute this performance advantage to our explicit modeling of self-evaluation signals and their seamless integration into policy learning through evaluation-guided strategy constraints. This synergy enables precise control over retrieval frequency and generation quality during complex multi-hop reasoning. Moreover, we observe that although s3 can be optimized with RL using the Gain-Beyond-RAG reward, its performance cannot be fully realized in a restricted retrieval environment. In larger-scale document retrieval scenarios with richer candidate pools, it would demonstrate greater advantages.

• **Importance of RL and Internal Alignment.** In multi-hop retrieval and reasoning tasks, both Search-R1 and Reflective RAG outperform Search-o1 and Self-RAG, underscoring the efficacy of reinforcement learning in optimizing multi-turn retrieval policies. Unlike Self-RAG, which inherits evaluation labels from external expert models (e.g., GPT), Reflective RAG relies exclusively on internal signals, fostering a fully self-contained and model-consistent evaluation mechanism. Crucially, our framework enforces evaluation-aware strategy constraints during the RL phase. This transforms evaluation signals from passive supervision into active constraints that directly steer policy optimization. Thus, Reflective RAG generates more reliable and coherent reasoning trajectories by capitalizing on high-utility evidence while suppressing irrelevant or misleading retrieved content.

4.4 Ablation Study

To assess the individual contributions of each training component in Reflective RAG, we conducted a comprehensive ablation study across all datasets

using the Qwen2.5-3B-Instruct backbone. The results are summarized in Table 2. 1) **Limitations of SFT.** While Supervised Fine-Tuning alone (SFT Only) yields noticeable improvements over the base model (Original), demonstrating its ability to enhance task adaptability, it still underperforms reinforcement learning-based training. Specifically, although SFT successfully instills basic competencies such as evaluation label prediction and output formatting, it proves insufficient for mastering the complex dynamics of multi-turn retrieval and information integration required for robust trajectory generation. 2) **Criticality of Constraints and Regularization.** When SFT is trained without evaluation label correction and trajectory reconstruction (No correction), the resulting model produces noisier evaluation signals, which in turn weakens the effectiveness of both strategy constraints and evaluation-aware KL regularization during RL. Moreover, removing strategy constraints during RL training (No Cons.) leads to a significant performance decline across most benchmarks, indicating that constraining the generation policy is essential for stabilizing multi-hop retrieval processes guided by self-evaluation. Furthermore, eliminating KL regularization on evaluation labels (No eval. KL) degrades performance. These findings confirm that label correction and trajectory reconstruction during SFT, together with strategy constraints and evaluation-aware KL regularization in RL, are critical for training stability. They ensure that the model accurately learns evaluation signals and generation strategies, ultimately producing more reliable reasoning trajectories.

4.5 Quantitative Analysis

Evaluation-guided Strategy Constraint during Inference. We analyze the effect of evaluation-

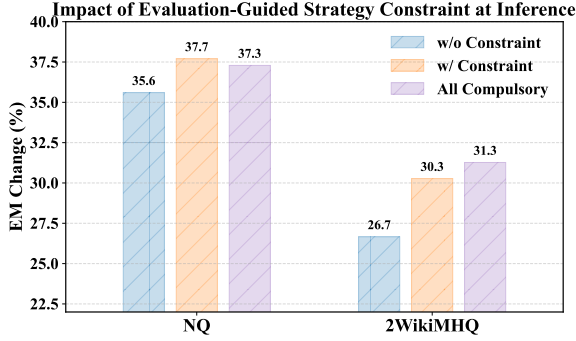


Figure 3: Performance of Qwen2.5-7B-Instruct on NQ and 2WikiMHQ under evaluation-guided strategy constraints.

Model	Method	NQ	2Wiki	HQA
Qwen 3B	Search-o1	1.67	1.97	1.69
	Search-R1	1.39	1.75	1.47
	Reflective RAG	1.95	1.96	1.96
Qwen 7B	Search-o1	2.44	3.56	2.85
	Search-R1	2.21	2.92	2.78
	Reflective RAG	1.60	1.83	1.74

Table 3: Comparison of average retrieval steps required by different methods on three datasets.

guided constraints at inference by comparing three strategies: (1) *No Constraint*, without additional inference constraints; (2) *With Constraint*, enforcing at least one extra retrieval only when the latest retrieval is evaluated as “Confusing” and the model attempts a final answer; (3) *All Compulsory*, performing an extra retrieval regardless of the evaluation outcome. As shown in Figure 3, selectively applying the constraint after encountering a “Confusing” signal (“With Constraint”) consistently improves performance across datasets. In contrast, the *All Compulsory* strategy yields smaller gains while incurring additional retrieval steps, and even slightly decreases performance on the NQ dataset. These results indicate that reasoning robustness is improved not by uniformly increasing retrievals, but by evaluation-guided strategy constraints, which leverage self-evaluation signals to selectively regulate inference-time retrieval and termination behaviors. More importantly, Appendix E presents a qualitative Case Study illustrating how evaluation-guided strategy constraints influence inference behavior in Reflective RAG.

Retrieval Frequency Analysis. Table 3 reports the average number of retrieval steps per trajectory for each method across datasets, evaluated with Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. For the 3B model, Reflective RAG issues more

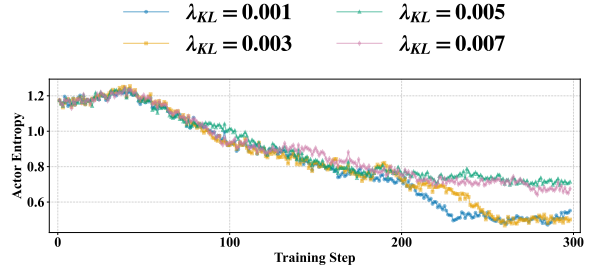


Figure 4: Evolution of actor entropy during training under varying evaluation label KL coefficients.

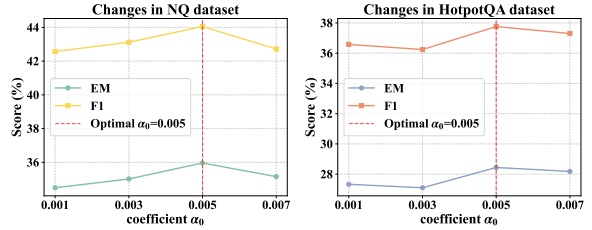


Figure 5: Impact of evaluation label KL coefficients on NQ and HotpotQA performance.

retrievals than Search-R1 and Search-o1, reflecting its evaluation-guided strategy constraints. When self-evaluation suggests that the current evidence is insufficient or unreliable, the model performs additional retrieval to support faithful reasoning. In contrast, for the 7B model, Reflective RAG performs fewer retrievals than the baselines, as its stronger internal evaluation allows it to use retrieved evidence more selectively and stops earlier with higher confidence. Overall, Reflective RAG neither blindly increases nor decreases retrieval frequency; instead, it adaptively calibrates retrieval via self-evaluation, adjusting behavior across model scales to achieve better performance.

Impact of Evaluation KL Coefficient. We vary the evaluation KL coefficient over $\{0.001, 0.003, 0.005, 0.007\}$. As shown in Figure 4, entropy evolves similarly across settings in the early phase, suggesting that the policy relies little on evaluation labels before alignment. In the later phase, smaller KL values (0.001, 0.003) lead entropy to plateau at low levels, as insufficient regularization makes the evaluation-label distribution rapidly conform to an under-aligned policy, leading to semantic drift and reduced adaptability. By contrast, larger KL values (0.005, 0.007) better preserve the reliability of evaluation signals, supporting sustained exploration with stable entropy. The setting $\alpha_0 = 0.005$ strikes the best balance between stability and exploration and achieves the highest final performance

across datasets (Figure 5). Overall, these results underscore the value of a moderately decayed KL constraint applied to evaluation signals, which maintains stable evaluation semantics early in training while gradually relaxing to enable adaptive strategy learning during RL.

5 Conclusion

In this work, we present Reflective RAG, an evaluation-aware agentic retrieval-augmented framework that enhances multi-turn reasoning via self-evaluation signals. Using self-corrected evaluation labels, incorporating evaluation-guided strategy constraints, and applying KL regularization on these labels, the framework improves reasoning stability and enables judicious use of retrieved information. Experiments on five knowledge-intensive QA benchmarks show that Reflective RAG consistently outperforms strong baselines, with ablations confirming the effectiveness of each component in stabilizing reasoning and improving generalization. This work advances more robust and consistent agentic RAG systems guided by internal self-evaluation mechanisms.

Limitations

Despite its effectiveness, Reflective RAG has several limitations that suggest directions for future research.

Training and evaluation on larger models. Our experiments primarily focus on small to medium-sized LLMs, which exhibit higher reasoning uncertainty, making deficiencies in retrieval utilization and reasoning stability more apparent. Reflective RAG shows clear improvements in this setting. However, systematic training and evaluation on larger models are necessary to verify the framework’s generalizability and effectiveness at higher model capacities.

Evaluation signals are not directly optimized as rewards. In Reflective RAG, self-evaluation signals are used to impose strategy constraints rather than as direct rewards. Preliminary attempts to design turn-level rewards based on evaluation labels led to reward hacking, where the model over-predicted “Useful” while avoiding “Redundant” and “Confusing”, thereby distorting the semantics of evaluation signals. Designing reward mechanisms that leverage self-evaluation without inducing such degenerate behaviors remains an open challenge.

Coarse-grained evaluation categories. The current reflection tagging mechanism classifies retrieved content into coarse-grained utility categories (e.g., Useful, Redundant, Confusing). While sufficient for high-level strategy constraints, these labels may overlook finer distinctions in evidence quality, relevance, or reliability. This limitation becomes particularly relevant in settings where multiple documents are retrieved at a single step, as different documents may vary in correctness or usefulness while being assigned a shared evaluation signal. Incorporating more fine-grained or structured evaluation signals could enable more nuanced strategy modulation, particularly in complex multi-hop reasoning scenarios or ambiguous retrieval tasks.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China under Grant (62306267), Zhejiang Province “Lingyan” Key R&D Project of China (No.2026C02A2002) and Zhejiang Province “Jianbing” Key R&D Project of China (No.2025C01010, No.2024C01034).

References

- Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia computer science*, 246:3781–3790.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, pages 1–30, Vienna, Austria. OpenReview.net.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. [Semantic parsing on Freebase from question-answer pairs](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.
- Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, and 1 others. 2025. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.
- Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. [A survey on rag meeting llms: Towards](#)

- retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24*, page 6491–6501, New York, NY, USA. Association for Computing Machinery.
- Michael Glass, Gaetano Rossiello, Md Faisal Mahub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. [Re2G: Retrieve, rerank, generate](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025a. s3: You don't need that much data to train a search agent via RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21599–21617, Suzhou, China. Association for Computational Linguistics.
- Yi Jiang, Sendong Zhao, Jianbo Li, Haochun Wang, and Bing Qin. 2025b. [GainRAG: Preference alignment in retrieval-augmented generation through gain signal synthesis](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10746–10757, Vienna, Austria. Association for Computational Linguistics.
- Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*, Honolulu, Hawaii, USA. JMLR.org.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.
- Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, and 1 others. 2024. Rag-ddr: Optimizing retrieval-augmented generation using differentiable data rewards.
- Jintao Liang, Gang Su, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. 2025. Reasoning rag via system 1 or system 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges. *arXiv preprint arXiv:2506.10408*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.
- Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou

- Yu, Weizhu Chen, and 1 others. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. [REPLUG: Retrieval-augmented black-box language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.
- Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.
- Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Jirong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training.
- Suyuchen Wang, Jinlin Wang, Xinyu Wang, Shiqi Li, Xiangru Tang, Sirui Hong, Xiao-Wen Chang, Chenglin Wu, and Bang Liu. 2025. Improving context fidelity via native retrieval-augmented reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21216–21229.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. 2025. [Reinforcing multi-turn reasoning in llm agents via turn-level reward design](#). *Preprint*, arXiv:2505.11821.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales.
- Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM Web Conference 2024*, pages 1362–1373.
- Yongxin Xu, Ruizhe Zhang, Xinke Jiang, Yujie Feng, Yuzhen Xiao, Xinyu Ma, Runchuan Zhu, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025. [Parenting: Optimizing knowledge selection of retrieval-augmented language models with parameter decoupling and tailored tuning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11643–11662, Vienna, Austria. Association for Computational Linguistics.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. [Augmentation-adapted retriever improves generalization of language models as generic plug-in](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2421–2436, Toronto, Canada. Association for Computational Linguistics.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, and 1 others. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025b. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, and 1 others. 2023. Siren’s song in the ai ocean: a survey on hallucination in large language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 2024. [LlamaFactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Percentile-based Evaluation Label Assignment

For each retrieval step, we assign an evaluation label based on the percentile distribution of its perplexity difference ΔPPL . Specifically, steps with $\Delta\text{PPL} > 0$ that fall within the top $\lambda_e\%$ of the positive ΔPPL distribution are labeled as “Useful”. Steps whose ΔPPL lies in the bottom $\lambda_e/2\%$ of the positive ΔPPL distribution or in the top $\lambda_e/2\%$

of the negative ΔPPL distribution are labeled as “Redundant”. Finally, steps with $\Delta\text{PPL} < 0$ that fall within the bottom $\lambda_e\%$ of the negative ΔPPL distribution are labeled as “Confusing”.

This percentile-based labeling strategy relies solely on the model’s internal generation probabilities and requires neither external expert models nor manual annotations, thereby enabling fully self-supervised correction of evaluation labels.

B GRPO Algorithm

GRPO trains the model policy $\pi_\theta(y | x)$ by computing group-relative advantages over a set of G rollout trajectories $\{y_i\}_{i=1}^G$. Incorporating our proposed modifications, the GRPO objective is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \left(\rho_{i,t} A_{i,t}, \text{clip}(\rho_{i,t}, 1 - \epsilon, 1 + \epsilon) A_{i,t} \right) - \beta D_{\text{KL}} \right], \quad (7)$$

where

$$\rho_{i,t} = \frac{\pi_\theta(y_{i,t} | x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} | x, y_{i,<t})}, \quad (8)$$

$$A_{i,t} = \frac{R_i - \mu_R}{\sigma_R}. \quad (9)$$

Here, $A_{i,t}$ denotes the normalized group-relative advantage, where μ_R and σ_R are the mean and standard deviation of rewards within the rollout group. The KL term D_{KL} corresponds to the evaluation-aware, linearly decayed KL regularization described in Equation 4.

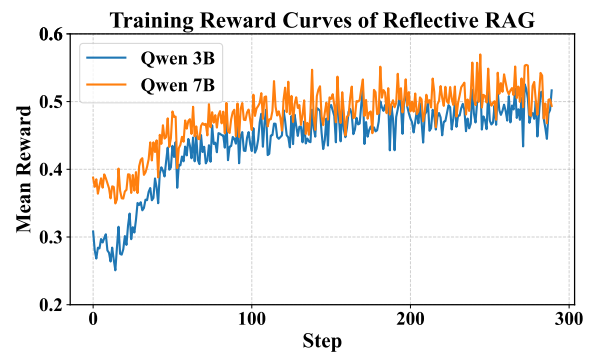


Figure 6: Training reward curves of Reflective RAG across different model scales during GRPO.

C Training

We visualize the reward curves of Reflective RAG during GRPO training for both Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. As shown

in Figure 6, the curves illustrate the training dynamics and demonstrate the stability of Reflective RAG’s reinforcement learning across different model scales.

D More Details on Experiment Setup

D.1 Implementation Details

We conduct supervised fine-tuning (SFT) and reinforcement learning (RL) using the LLaMA-Factory (Zheng et al., 2024) and verl (Sheng et al., 2025) frameworks, respectively. All experiments are performed on a single node with four 80GB GPUs. For SFT, the model is trained on a curated dataset of 22k trajectories, combining filtered samples from NQ, HotpotQA, and TriviaQA (Joshi et al., 2017). We use AdamW optimizer (Loshchilov and Hutter, 2017) with an initial learning rate of $2e-6$, a warmup ratio of 0.1, and a total batch size of 256 for one epoch to avoid overfitting. For RL, NQ and HotpotQA are merged into a unified training set. Training uses a total batch size of 256, mini-batch size of 128, micro-batch size of 20, and a maximum sequence length of 4096. The learning rate is $5e-7$ with a warmup ratio of 0.285, trained for $T_{\text{train}} = 300$ steps with a clip ratio $\epsilon = 0.2$. During rollout, vLLM is employed to sample five response trajectories per prompt. Checkpoints are saved every 60 steps, and the model used for evaluation is either the final checkpoint or the last checkpoint before any reward collapse.

D.2 Prompt Templates

During SFT data collection, after the evaluation labels are corrected, each label is paired with a prompt that guides subsequent reasoning, as illustrated in Figure 7. Figure 8 shows the instruction prompt used during both training and evaluation.

E Case Study

As illustrated in Figure 9 and Figure 10, we present two examples from the HotpotQA dataset showing how the baseline without strategy constraints attempts to answer under high uncertainty, whereas Reflective RAG enforces additional retrieval via strategy constraints to ensure grounded reasoning.

Prompts Guiding Reasoning for Corrected SFT Labels

Useful: The previous <information> may be useful, so I will incorporate it into my reasoning and continue reasoning.

Redundant: The previous <information> may be redundant or already known, so I will leave it alone and continue reasoning.

Confusing: The previous <information> may be confusing or misleading, so I will ignore it and continue reasoning.

Figure 7: Prompts for each corrected SFT label guiding subsequent reasoning.

Prompt Templates for Reflective RAG

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. Before using the retrieved information, evaluate its relevance to the final answer using the reflection token: Useful, Redundant, or Confusing — and output that token inside <evaluation></evaluation>.

- Useful: The retrieved information directly improves the reasoning process or provides valuable context to the final answer.
- Redundant: The retrieved information is already known or does not contribute new insights to the reasoning process.
- Confusing: The retrieved information may mislead the reasoning process.

When you mark an <information> block as Confusing, you should ignore its content for now and continue reasoning based on the reliable context accumulated so far. You can search as many times as you want, but remember to evaluate each new piece of information before using it. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> xxx </answer>. You should output in the following loop format:

```
"<think>\nxxx\n</think>\n<search>\nxxx\n</search>\n<information>\nxxx\n</information>\n<evaluation>\nxxx\n</evaluation>\n...\n<think>\nxxx\n</think>\n<search>\nxxx\n</search>\n<information>\nxxx\n</information>\n<evaluation>\nxxx\n</evaluation>\n<think>\nxxx\n</think>\n<answer>\nxxx\n</answer>".
```

Make sure the output format is correct. Question: {question}

Figure 8: Template used for training and inference.

Inference with Constraint (Evaluation-Guided strategy Constraints)

Question: Lawrence Turman had produced a film, with a character named "Johnny 5" that had a follow up sequel released in what year?

Golden answers: ['1988']

<think>Lawrence Turman produced a film with a character named "Johnny 5". This film had a sequel. I need to find the release year of the sequel.</think>

<search>When was the sequel to the film with character "Johnny 5" released?</search>

<information>Doc 1(Title: "Johnny English Reborn") Kingdom on 7 October 2011, and topped the country's box office for the next two weekends, before being dethroned by ""Paranormal Activity 3"". It was later released in North America on 21 October 2011. A sequel to the film, ""Johnny English Strikes Again"" was released in October 2018. Johnny English has been hiding in Tibet following a botched mission in Mozambique (during which he failed to protect the newly-elected president) when he is summoned by MI7. Under his new boss Pamela Thornton, codename ""Pegasus"", he is put on a mission to investigate a plot to assassinate the Chinese Premier during</information>

<evaluation>Confusing</evaluation>

<think>The information provided seems confusing as it talks about "Johnny English" series instead of "Johnny 5". I don't have enough relevant information to determine the release year of the sequel to the film with character "Johnny 5".</think>

<answer>1986</answer> × (attempted answer under high internal uncertainty; blocked by strategy constraint and followed by forced retrieval)

<search>What year was the sequel to the film with character Johnny 5 released?</search>

<information>Doc 1(Title: "Short Circuit 2") Short Circuit 2 Short Circuit 2 is a 1988 US comic science fiction film, the sequel to the 1986 film ""Short Circuit"". It was directed by Kenneth Johnson, and starred Fisher Stevens as Ben Jahveri, Michael McKean as Fred Ritter, Cynthia Gibb as Sandy Banatoni, and Tim Blaney as the voice of Johnny 5 (the main character – a friendly, naive, self-aware robot). After Nova Robotics goes bankrupt after the ""destruction"" of one of their robot prototypes by Captain Skroeder, one of their programmers, Benjamin Jahveri (Fisher Stevens), starts up his own business, ""Titanic Toy Corporation"", that specializes in making</information>

<evaluation>Useful</evaluation>

<think>The sequel to the film with character Johnny 5, which is actually called Short Circuit 2, was released in 1988.</think>

<answer>1988</answer> ✓

Analysis: In this case, after the initial retrieval is evaluated as “Confusing”, the model correctly recognizes that the evidence is insufficient. Without constraints, it may still attempt a plausible answer based on partial internal associations, resulting in premature conclusions without support from retrieved documents. The evaluation-guided constraint prevents this early answer and enforces an additional retrieval step, providing the necessary evidence for a fully grounded and correct inference.

Figure 9: A case of Reflective RAG.

Inference with Constraint (Evaluation-Guided strategy Constraints)

Question: Who was the main vocal of "Set the Fire to the Third Bar"?

Golden answers: ['Gary Lightbody']

<think>The question is about the main vocalist of the song "Set the Fire to the Third Bar". I need to find out who the main vocalist is for this song.**</think>**

<search>Who was the main vocal of "Set the Fire to the Third Bar"?**</search>**

<information>Doc 1(Title: "...And Then There Were Three...") band instead settled on a set of lyrics that Collins offered during down time while mixing at Trident Studios that brought in a different melody and more harmonies. ""Say It's Alright Joe"", written by Rutherford and the penultimate track recorded for the album, is a torch song about an alcoholic who goes into a drunken stupor. The guitarist intended the track to be a ""piss-take on the Dean Martin 'set 'em up Joe' alcoholic style'"" , but thought it was not going to work until Banks added his keyboard overdubs and the band started mixing the track, at which point it**</information>**

<evaluation>Confusing**</evaluation>**

<think>The information provided is about another song and band, which is not helpful. I need to find the main vocalist for "Set the Fire to the Third Bar".**</think>**

<answer>Nick Mason**</answer>** × (attempted answer under high internal uncertainty; blocked by strategy constraint and followed by forced retrieval)

<search>Main vocalist of Set the Fire to the Third Bar**</search>**

<information>Doc 1(Title: "Set the Fire to the Third Bar") in your warm arms"). After their long journey to meet each ("After I have travelled so far") they would settle down in front of an electric bar fire ("We'd set the fire to the third bar") and enjoy being together and warm. The song was recorded at Grouse Lodge Studios, Westmeath, and Apollo Studios, Dublin. The song has been remixed by Anu Pillai of the band Freeform Five, a friend of Gary Lightbody and Tom Simpson. On 12 November 2009, Snow Patrol performed a version of ""Set the Fire to the Third Bar"" with former Girls Aloud singer Cheryl for**</information>**

<evaluation>Useful**</evaluation>**

<think>The song "Set the Fire to the Third Bar" is performed by Snow Patrol. The main vocalist of Snow Patrol is Gary Lightbody.**</think>**

<answer>Gary Lightbody**</answer>** ✓

Analysis: This example illustrates another failure mode under “Confusing” evidence. After recognizing that the retrieved document is irrelevant, the model may still attempt to generate an answer under high internal uncertainty, producing an arbitrary and incorrect response. Unlike the previous case, the output is not a plausible inference but an ungrounded guess. The evaluation-guided constraint mitigates such uncertainty-driven hallucinations by enforcing additional retrieval, enabling the model to ground its reasoning once “Useful” evidence is obtained.

Figure 10: A case of Reflective RAG.