Adversarial Yet Cooperative: Multi-Perspective Reasoning in Retrieved-Augmented Language Models

Can Xu[1,2], Lingyong Yan[2], Jiayi Wu[1], Haosen Wang[3], Shuaiqiang Wang[2], Yuchen Li[2], Jizhou Huang[2], Dawei Yin [2], Xiang Li[1],

[1]East China Normal University, [2]Baidu Inc., [3]Southeast University

Correspondence: Xiang Li xiangli@dase.ecnu.edu.cn

Abstract

Recent advances in synergizing large reasoning models (LRMs) with retrieval-augmented generation (RAG) have shown promising results, yet two critical challenges remain: (1) reasoning models typically operate from a single, unchallenged perspective, limiting their ability to conduct deep, self-correcting reasoning over external documents, and (2) existing training paradigms rely excessively on outcome-oriented rewards, which provide insufficient signal for shaping the complex, multi-step reasoning process. To address these issues, we propose an Reasoner-Verifier framework named Adversarial Reasoning RAG (ARR). The Reasoner and Verifier engage in reasoning on retrieved evidence and critiquing each other's logic while being guided by process-aware advantage that requires no external scoring model. This reward combines explicit observational signals with internal model uncertainty to jointly optimize reasoning fidelity and verification rigor. Experiments on multiple benchmarks demonstrate the effectiveness of our method. Our code is available at link.

1 Introduction

Large language models (LLMs) endowed with step-by-step reasoning capabilities have achieved remarkable success in complex question answering, especially when augmented with external knowledge through retrieval-augmented generation (RAG) (Li et al., 2025c,b; Feng et al., 2025; Wang et al., 2025). Different from previous RAG methods that focus on retrieval optimization and component-based architectural design, recent efforts have been made on post-training LLM agents (Jin et al., 2025a; Li et al., 2025a) integrated with search tools.

Despite the effectiveness, current RAG mainly adopts a monologic reasoning architecture, where only one single LLM-based agent reasons and interacts with search engines. However, when retrieved documents are partial, conflicting or misleading, the single-view reasoning may amplify errors rather than mitigate them. Prior efforts address this challenge by incorporating self-verification process (He et al., 2025; Fu et al., 2025). However, such self-critique paradigm also suffers from the single-view architecture, as many studies (Xu et al., 2024; Wu et al., 2025; Zhang et al., 2025) show that LLMs struggle to identify their own logical flaws.

Moreover, in order to train the agentic RAG system, most existing methods optimize the RL framework using outcome-oriented, task-level rewards (e.g., accuracy or format correctness). Such rewards assign uniform reward to tokens within a sequence based on the final correctness, lacking supervision for the intermediate process. Unlike self-contained trajectories in mathematical domains, the correctness in RAG system depends not only on reasoning quality, but on external factors beyond the agent's control, such as the precision of retrieval engine, the consistency of external documents, and the presence of conflicting evidence. Therefore, outcome-based rewards cannot distinguish between a correct answer derived through sound logic and the one produced by lucky guesswork, nor can they penalize plausible but flawed reasoning that happens to yield a wrong answer.

To tackle these challenges, we propose **ARR** (**A**dversarial **R**easoning **R**AG), a multi-perspective framework that explicitly decouples reasoning and verification into separate perspectives, handled by a reasoner agent and a verifier agent, respectively. And we formalize such interactive process as an adversarial yet collaborative dialogue between them:

Adversarial yet cooperative interaction: The two agents should challenge each other not for winning the debate, but for a shared objective. Critiques should be justified and evidence-grounded.

Process-aware learning: The two agents are rewarded not only for correct final answers, but also for high-quality interactive process between

Findings of the Association for Computational Linguistics: ACL 2026, pages 37469–37480

them (e.g., logical coherence, evidence utilization, and uncertainty reduction).

To this end, we introduce an *adversarial outcome reward* and a *process-aware advantage* into the co-evolving process of both agents. (1). The adversarial outcome reward encourages agents to compete for higher correctness, ensuring that the consensus is driven by rigorous debate rather than blind agreement. (2). The process-aware advantage is a token-level advantage for the verifier, which is driven by a core insight: high-quality reasoning in RAG should mirror the reduction of uncertainty and semantic entropy. With the proposal of search queries and the accumulation of evidences, the agent moves from an initial state of confusion to a state of crystallization. Based on this guiding principle, the process-aware advantage assesses the soundness of response, the clarity of verification, and the impact on the reasoner's cognitive state. By monitoring the evolution of reasoner policy entropy, we reward verifier's feedback that is confident, evidence-grounded, and steers the reasoner from high-entropy exploration to low-entropy convergence, thereby aligning the optimization with the information gain.

In summary, our main contributions are:

(1) We propose ARR, where reasoner and verifier engage in adversarial yet cooperative dialogue.

(2) We propose adversarial outcome reward to promote rigorous debate between agents, which encourages agents to compete for higher correctness.

(3) We design the token-level process-aware advantage. By modeling reasoning progress as the reduction of uncertainty, we reward trustworthy and evidence-grounded verifier feedback that effectively steers the reasoner toward better reasoning.

2 Related Work

2.1 Reward Design and Process Supervision

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a powerful approach to enhance the reasoning capabilities of LLMs. For example, Pass@k (Chen et al., 2025b) reveals that outcome rewards provides limited learning signals for tasks that are either overly simple or difficult, and fail to discriminate between effective and ineffective process within the reasoning trace. It leverages pass@k performance as the replacement for outcome only rewards. DAPO (Yu et al., 2025) introduces dynamic sampling to filter out samples where model consistently succeeds or fails. In reasoning RAG scenrario, Atom-Searcher (Deng et al., 2025) introduces reasoning reward model to provide process signal additional to outcome reward. WebSeer (He et al., 2025) introduces F1-score as the intermediate-step verification signal to guide the exploration process of search agent.

2.2 LRMs Synergizied with RAG

Recent advances in RAG systems include the integration of search tools and LRMs, which significantly improve the capabilities for complex and multi-step reasoning and searching. The representative methods Search-R1 (Jin et al., 2025a) and R1-Searcher (Song et al., 2025) train models to automatically derive reasoning through multi-turn searching. DeepResearcher (Zheng et al., 2025) further includes web search agent into agentic reasoning RAG. Existing methods are primarily built upon single-agent frameworks, leaving a gap in exploration from multi-perspective interactions.

3 Preliminary

3.1 Task Formulation

An ideal agentic reasoning RAG system should go beyond the search-retrieve-answer pipeline and possess high-order capabilities, including:

Critical reasoning: the capability to assess the reliability of external evidence and detect logical flaws in reasoning traces;

Grounded generation: the ability to anchor reasoning in verifiable evidence, and to revise conclusions when support is insufficient;

Iterative refinement: the ability to enhance reasoning quality through self- and peer-assessment, balancing both accuracy and process behaviors.

Current agentic RAG systems, however, remain constrained by monologic architectures and optimization objectives that rely predominantly on scaler outcome rewards. To bridge this gap, We propose ARR, a multi-perspective reasoning framework in which two agents learn to reason not as single voices, but through interaction of different viewpoints. Formally, we model the system as a multi-agent Markov Decision Process (MDP), defined as the tuple $(\mathcal{S}^\alpha, \mathcal{A}^\alpha, \mathcal{P}^\alpha, \mathcal{R}^\alpha)$. Let $\alpha \in \{\text{r,v}\}$ index the **Reasoner** (r) and the **Verifier** (v), interacting in an environment that includes search engine and document corpus D. Given a query q, the agent behavior is governed by the policy model π_θ^α. State $s_t^\alpha \in \mathcal{S}^\alpha$ refers to previous histories and external context by the agent other than α, and $a_t^\alpha \in \mathcal{A}^\alpha$ is

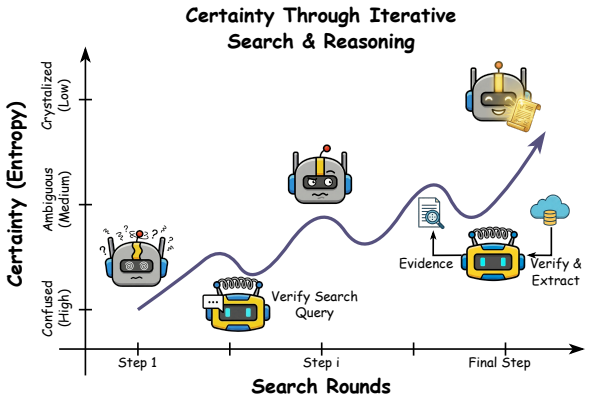

Figure 1: Ideal agent certainty through iterative search and reasoning

the action generated by π_θ^α from its action space at turn t: $a_t^\alpha = \pi_\theta^\alpha(s_t^\alpha)$. Notably, distinct from token-level MDPs, we define the action space $\mathcal{A}^\alpha$ at the semantic step level. An action a_t^α is a sequence of tokens representing a complete move. Take Search-R1 for an example, the action space $\mathcal{A}^\alpha$ = {`think`, `search`, `answer`}. A complete trace τ of n interaction steps is denoted as $\tau = (s_1^{\mathrm{r}}, a_1^{\mathrm{r}}, s_1^{\mathrm{v}}, a_1^{\mathrm{v}}, ..., s_n^{\mathrm{r}}, a_n^{\mathrm{r}}, s_n^{\mathrm{v}}, a_n^{\mathrm{v}})$.

3.2 Entropy Pattern Analysis

Generally, RLVR for LLMs often involves the trade-off between policy entropy and performance (Cui et al., 2025). In the context of RAG, the reasoning process can be regarded as the dynamic evolution of cognitive states driven by external knowledge management. As illustrated in Figure 1, an ideal reasoning trajectory exhibits three stages. (1) *Initial uncertainty*: the agent begins in a confused state, where high policy entropy reflects the lack of knowledge and exploration of search queries. (2) *Evidence integration*: the agent assimilates retrieval results and converges towards final answer. (3) *Crystallization*: the agent has sufficient evidence and generates a well-supported conclusion. To formalize this intuition, we present the following proposition.

Proposition 1. *In an ideal agentic RAG system, as relevant information is retrieved, both the uncertainty of agent and the policy entropy monotonically decrease.*

Proof. Consider an agentic RAG system with single agent (e.g. Search-R1). Let Y denotes the ground-truth answer. Given a user query q, the state s_{t+1} is the union of prior state s_t, action a_t and retrieved document d_t. To ensure rigor, we introduce two assumptions.

Assumption 1: The agent acts to maximize the expected information gain and issues search queries intended to retrieve relevant documents.

Assumption 2: The retrieved documents d_t provides a positive information gain regarding Y.

(1). Monotonic Decrease of Answer Uncertainty: The uncertainty of Y with document d_t is quantified by the conditional mutual information:

$$I(Y; d_t|s_t, a_t) = H(Y|s_t, a_t) - H(Y|s_{t+1}). \quad (1)$$

By definition, the updated state is $s_{t+1} = [s_t, a_t, d_t]$. Since a_t is generated based on state s_t, we have the Markov property $H(Y|s_t, a_t) \approx H(Y|s_t)$. Substituting these into Eq 1, we have: $H(Y|s_{t+1}) \approx H(Y|s_t) - I(Y; d_t|s_t, a_t)$. Since mutual information is non-negative, and the retriever provides relevant information, we obtain:

$$H(Y|s_{t+1}) \leq H(Y|s_t). \quad (2)$$

This indicates that remaining uncertainty of the ground-truth is non-increasing with the accumulation of retrieved documents.

(2). Convergence of Policy Entropy: Next, we discuss the entropy of action a_t, noted as $H(a_t|s_t)$. We decompose it using the definition of mutual information between action a_t and ground-truth Y:

$$H(a_t|s_t) = I(a_t; Y|s_t) + H(a_t|Y, s_t). \quad (3)$$

Then, we analyze the two terms on the right side of Eq 3. First, the mutual information is defined as $I(a_t; Y|s_t) = H(Y|s_t) - H(Y|a_t, s_t)$. Thereby, $I(a_t; Y|s_t) \leq H(Y|s_t)$ holds. Second, the term $H(a_t|Y, s_t)$ represents the uncertainty of agent's action given that the ground truth is known. Following Assumption 2, as the training progresses, given the ground-truth answer, the agent would gradually come to a deterministic output, i.e., $H(a_t|Y, s_t) \approx 0$. Therefore, we have the upper bound for the policy entropy:

$$H(a_t|s_t) \leq H(Y|s_t) + H(a_t|Y, s_t). \quad (4)$$

Since $H(Y|s_t)$ is monotonically decreasing, and $H(a_t|s_t)$ follows the upper bound of $H(Y|s_t)$. This illustrates that as the agent accumulates evidence, its reasoning process naturally converges from exploration to exploitation. □

To empirically validate this theoretical proposition, we conduct a statistical analysis of the policy

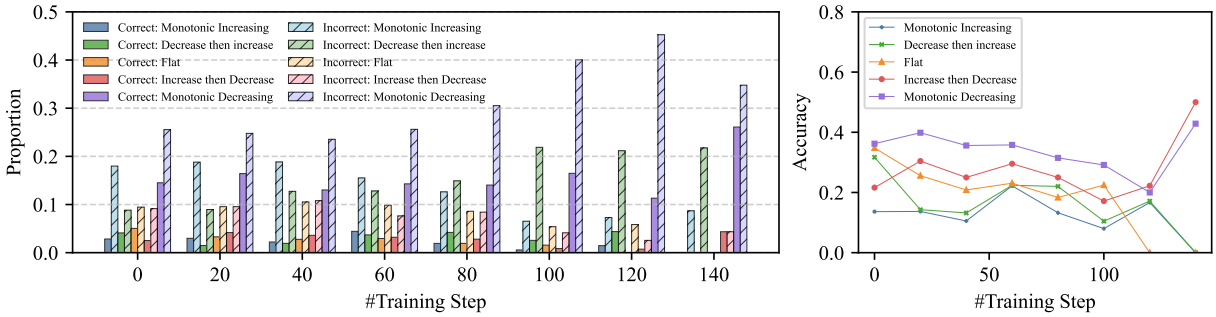

Figure 2: Statistical analysis of policy entropy pattern in Search-R1 trajectories. The y-axis of the **left subplots** denotes the proportion of trajectories exhibiting specific pattern in all multi-turn (≥ 3) samples. The y-axis of the **right subplots** represents the average accuracy of samples grouped by their pattens.

entropy evolution during the training process of Search-R1 (Jin et al., 2025a) in Figure 2[1]. Specifically, we focus on agent trajectories containing at least three search & reasoning turns and aggregate the statistics every 20 training steps. The action entropy is $H_{a_t} = \frac{1}{|a_t|}\sum_{j=1}^{|a_t^v|} H(\pi_\theta(a_{t,j}|s_t, a_{t,<j}))$, where $a_t \in \{\texttt{think}\}$ and $|\cdot|$ measures the sequence length of action a_t. Specifically, the trend between action a_{t+1} and a_t is *Increase*, if $\Delta H_{a_{t+1}} > \delta$; *Decrease*, if $\Delta H_{a_{t+1}} < \text{-}\delta$; *Flat*, otherwise. Here, $\Delta H_{a_{t+1}} = H_{a_{t+1}} - H_{a_t}$ and δ is the threshold which accounts for minor fluctuations during reasoning. For each of them, we track the average token entropy of last three turns and categorize its evolution trend into five patterns: *Monotonic Increasing* (I), *Decrease-then-Increase* (DI), *Flat* (F), *Increase-then-Decrease* (ID), and *Monotonic Decreasing* (D). We define a mapping function f_e: $R^n \rightarrow$ {D, ID, F, DI, I}. Figure 2 leads to two primary observations:

Correlation with Correctness: There is a positive correlation between the *Monotonic Decreasing* entropy pattern and model's accuracy. This suggests that effective reasoning is often accompanied by a progressive resolution of uncertainty.

Evolution of Exploration: Throughout training, there is a notable rise in the proportion of samples exhibiting an overall reduction in policy entropy (i.e., *Increase-then-Decrease*, and *Monotonic Decreasing*). Quantitatively, for the Qwen2.5-3B backbone, the proportion rises from 51.74% in the early phase to 69.57% in the late training phase. This suggests that the model learns to narrow down search space and converge on valid solutions as its multi-turn exploration capability deepens.

These observations support the premise that successful reasoning in RAG systems is intrinsically characterized by the progressive resolution of policy uncertainty. Collectively, this theoretical insight and empirical observation provide a robust foundation for the process reward design within our proposed multi-agent framework.

[1]We only show results on Qwen2.5-3B here, and more results are shown in Appendix.

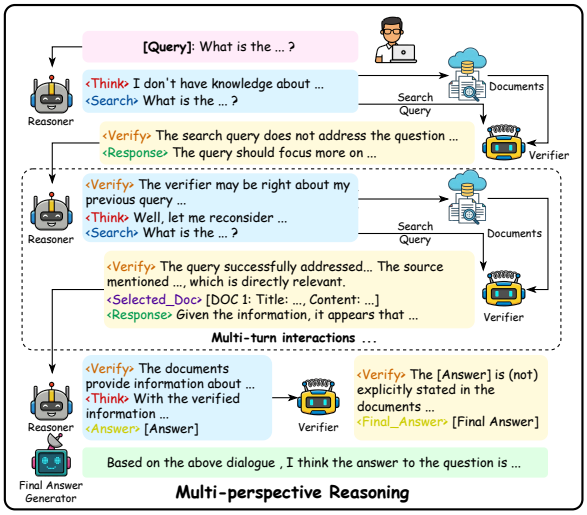

Figure 3: Multi-perspective reasoning of ARR.

4 Methods

4.1 Multi-perspective Reasoning

Building upon the formulation above, ARR performs multi-perspective reasoning and verification through an iterative dialogue between two agents:

Reasoner takes the lead in exploration. It formulates search queries, retrieves documents, constructs step-by-step reasoning, and proposes candidate answers.

Verifier serves as a critical partner. It checks the relevance and credibility of search queries and retrieved documents, identifies logical gaps or unsupported claims in reasoning, and performs validation

of the answers proposed by the reasoner.

For the reasoner, the action space is defined as $\mathcal{A}^{\text{r}}$ = {think, search, verify, answer}, a complete reasoning step is (think, search, [feedback], verify), and the final step is (think, answer).

• think: the segment of reasoning grounded in the given query q and retrieved evidence;

• search: search queries issued when external knowledge is deemed necessary;

• [feedback]: the feedback provided by the verifier, including supporting evidence or critiques;

• verify: self-assessment of the reliability and sufficiency of external knowledge information.

• answer: the answer when the reasoner thinks reasoning is complete and well-supported. Correspondingly, the verifier operates within the action space $\mathcal{A}^{\text{v}}$ = {verify, selected_doc, response, final_answer}. A complete verification step is ([information], verify, selected_doc, response), and the final step is ([information], verify, final_answer).

• [information]: search queries by the reasoner associated with retrieved documents or the reasoner's answer once it finishes reasoning;

• verify: verification on validity of queries, document relevance, and logical soundness;

• selected_doc: curated documents (e.g. Doc n) that directly support or refute the claim;

• response: explicit feedback to the reasoner, comprising either supporting evidence for valid queries or constructive critiques with justification for flawed ones.

• final_answer: the final conclusion after verifying all the evidence and the reasoner's answer.

Notably, the reasoner has a built-in verify stage within each step, allowing it to critically assess the reliability of retrieved evidence before coming to a conclusion. The verifier is instructed to return the most relevant source passage in selected_doc, rather than returning all retrieved documents to the reasoner without indiscriminately. This prevents information overload while guaranteeing feedback retains traceable and verifiable evidence. Together, these mechanisms help form a balanced adversarial dialogue, where neither agent dominates, and reasoning quality emerges from their structured interactions.

Following the iterative dialogue, we concatenate the full interaction history τ into a unified prompt, which is then fed to the final answer generator. Note that it employs the same policy model as the reasoner. By explicitly synthesizing insights from both perspectives, we obtain a more robust and well-grounded final answer. Detailed prompts for all agents are provided in Appendix A.3.

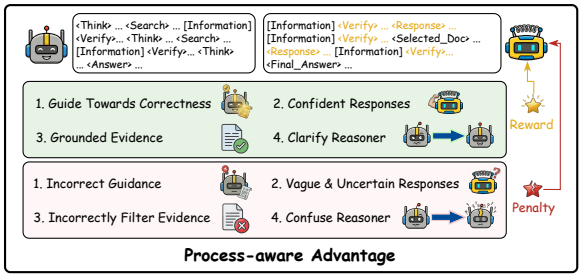

Figure 4: Process-aware advantage of ARR.

4.2 Multi-perspective Optimization

To overcome the limitations of sparse outcome-based supervision and to explicitly promote constructive adversarial interactions, we propose a multi-perspective reward design that disentangles final correctness from process fidelity. This design ensures that agents are rewarded not only for generating correct answers but also for engaging in high-quality and evidence-grounded dialogue. Our reward scheme consists of two components: adversarial outcome reward and process-aware advantage for the verifier.

Adversarial Outcome Rewards In consistent with the adversarial yet cooperative dialogue design, our outcome reward explicitly promotes effective adversarial engagement by rewarding agents not just for its' correctness, but for outperforming their counterpart. Formally, each agent α receives an outcome reward composed of two terms:

$$r^{\alpha} = \text{F1}(y^{\alpha}, y_{\text{gold}}) + \lambda \cdot \max\left[\texttt{bin}(r^{\alpha}_{\text{out}} - r^{\bar{\alpha}}_{\text{out}}), 0\right], \tag{5}$$

where $\bar{\alpha}$ denotes the counterpart agent, y^{α} denotes the answer by agent α, and y_{gold} is the ground truth. The operator $\texttt{bin}(\cdot)$ discretizes the range of F1 score into n buckets, filtering minor differences between answers by both agents. Thus, both agents are rewarded not only for correctness, but also for better performance than the other agent. Such reward also helps increase the discrimination of rewards within a group in Group Relative Policy Optimization (GRPO), particularly for tasks of moderate difficulty.

Process-aware Advantage While outcome reward drives both policy models towards accurate answers, they provide limited signal regarding how

correctness is achieved. To address this, we introduce a token-level process-aware advantage A^{v}_{proc} for the verifier, which encourages trustworthy and evidence-grounded response that steer the reasoner toward better reasoning. Formally, we define:

$$A^{\text{v}}_{proc} = \text{F1}(y^{\text{r}}, y_{\text{gold}}) \cdot A_{\text{clarity}} \cdot A_{\text{impact}}, \quad (6)$$

where the three terms each encodes correctness, clarity, and behavior impact, respectively.

(1) Answer Correctness The reasoner's final F1 score serves as a necessary condition: only when the dialogue yields a correct answer does the verifier receive full process credit. This prevents rewarding the verifier for critiques that lead reasoning toward wrong conclusions.

(2) Verifier Clarity

$$A_{\text{clarity}} = \exp(-H^{\text{v}}_{a_t}) \cdot \mathbb{I}[y_{\text{gold}} \text{ in } d_t] \cdot (2\mathbb{I}[y_{\text{gold}} \text{ in } a^{\text{v}}_t] - 1), \quad (7)$$

where $d_t \in D$ is the retrieved documents at step t, and a^{v}_t is the verifier's action in $\mathcal{A}^{\text{v}}_{\text{sub}}$ (e.g., {`verify`, `response`}). Here, $H^{\text{v}}_{a_t}$ denotes the average policy entropy of action a^{v}_t: $H^{\text{v}}_{a_t} = \frac{1}{|a^{\text{v}}_t|}\sum_{j=1}^{|a^{\text{v}}_t|} H(\pi^{\text{v}}_\theta(a^{\text{v}}_{t,j}|s^{\text{v}}_t, a^{\text{v}}_{t,<j}))$ In the first term, lower entropy means higher semantic certainty. We encourage confident and decisive critiques. The second term ensures that the verifier only receives credit or punishment when the final answer is actually supported by the retrieved documents, mitigating bias from imperfect retrieval. The third term penalizes responses that filter correct answers, thereby promoting faithfulness.

(3) Behavior Impact Most critically, we quantify how the verifier's feedback influences subsequent reasoning. Let $H^{\text{r}}_{a_t}$ denote the average token-level entropy of reasoner's action a^{r}_t. Over a dialogue of n interaction steps, we analyze the entropy trend in the last three steps and classify it into one of the five patterns defined in Section 3.2. We then assign a score to each pattern: $\text{score}(p) = \{\text{D} \rightarrow 1.0,\ \text{ID} \rightarrow 0.8,\ \text{F} \rightarrow 0.6,\ \text{DI} \rightarrow 0.4,\ \text{I} \rightarrow 0.2\}$ Finally, the impact is then:

$$\mathcal{A}_{\text{impact}} = \frac{1}{|\mathcal{A}^{\text{r}}_{\text{sub}}|} \sum_{j=1}^{|\mathcal{A}^{\text{r}}_{\text{sub}}|} \text{score}\left(f_e([H^{\text{r}}_{a_1}, ..., H^{\text{r}}_{a_n}])\right), \quad (8)$$

where $\mathcal{A}^{\text{r}}_{\text{sub}}$ is the set of reasoner action subjected to monitoring. This term incentivizes the verifier to provide feedback that steers the reasoner toward low-entropy and decisive reasoning.

4.2.1 Policy Optimization

We optimize both agents using Group Relative Policy Optimization (GRPO), which normalizes advantages within a group of rollouts and incorporates a reference model for KL regularization. For each query q, we sample G traces $\{\tau_i\}_{i=1}^G$ and calculate the outcome reward $\{r^\alpha_i\}_{i=1}^G$. The token-level advantage for t-th token in trace i is first computed as: $A^\alpha_{i,t} = \frac{r^\alpha_i - \text{mean}(r^\alpha_1, r^\alpha_2, ..., r^\alpha_G)}{\text{std}(r^\alpha_1, r^\alpha_2, ..., r^\alpha_G)}$ For the reasoner, the final advantage is $\hat{A}^{\text{r}}_{i,t} = A^{\text{r}}_{i,t}$. For the verifier, the process-aware advantage is added:

$$\hat{A}^{\text{v}}_{i,t} = A^{\text{v}}_{i,t} + \mathbb{I}(y_{i,t} \in a_t \wedge a_t = \mathcal{A}^{\text{v}}_{sub}) \cdot A^{\text{v}}_{proc} \quad (9)$$

ensuring $A^{\text{v}}_{\text{proc}}$ is only added to tokens belonging to the verifier's critique sections. Therefore, the policy model is optimized by maximizing:

$$\mathcal{J}^\alpha_{GRPO}(\theta) = \mathbb{E}_{\substack{q,\{y_{i,t}\}_{i=1}^G \sim \\ \pi^\alpha_{\text{old}}(\cdot|x;\mathcal{R})}} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|} \min\left(r^\alpha_i(\theta)\hat{A}^\alpha_{i,t}, \text{clip}(r^\alpha_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^\alpha_{i,t} \right) - \beta\mathbb{D}_{KL}\left[\pi^\alpha_\theta || \pi^\alpha_{\text{ref}}\right] \Bigg], \quad (10)$$

where $r^\alpha_i(\theta) = \frac{\pi^\alpha_\theta(y_{i,t}|q)}{\pi^\alpha_{\text{old}}(y_{i,t}|q)}$ and π^α_{ref} is the reference model. Following common practices in this field, tokens not generated by the policy model π^α_{old} will be masked in the loss calculation.

5 Experiments

5.1 Setup

Datasets & Metrics We conduct evaluation on diverse QA benchmarks. Our method and all baselines are trained on NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018). Following previous studies (Zheng et al., 2025), we randomly sample 512 examples from the development set of NQ, HotpotQA, TriviaQA (Joshi et al., 2017), 2WikiMultiHopQA (Ho et al., 2020), PopQA (Mallen et al., 2023), and MuSiQue (Trivedi et al., 2022), as well as all 125 samples from Bamboogle (Press et al., 2023). We adopt Exact Match (EM) and F1-score for comparison.

Baselines We compare our method against several baselines for reasoning and RAG in question answering, including CoT (Wei et al., 2022), RAG (Lewis et al., 2020), Search-R1 (Jin et al.,

Method	General QA						Multi-hop QA								Average	
	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Musique		Bamboogle			
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen2.5-3B Instruct																
CoT	2.5	8.7	7.3	13.2	7.4	13.7	3.4	12.5	2.1	12.4	0.2	3.2	0.0	0.0	3.27	9.1
RAG	33.7	39.6	51.5	58.4	36.5	44.2	22.1	30.4	21.6	29.9	7.2	13.6	60	12.3	25.5	32.6
Search-R1	38.7	45.9	55.2	66.7	40.0	50.2	31.4	38.3	32.4	41.2	10.2	20.8	18.9	25.6	32.4	41.2
- *pass@2*	42.9	-	64.5	-	42.2	-	34.1	-	35.0	-	12.8	-	25.6	-	36.7	-
ReSearch	39.5	46.7	59.3	67.8	41.4	49.9	32.9	38.6	33.3	42.5	11.4	21.0	24.7	31.2	34.6	42.5
WebSeer	39.0	46.3	58.5	67.1	40.7	50.6	35.3	44.4	34.8	43.6	11.5	21.7	24.3	32.5	34.8	43.7
ARR	**43.7**	**53.2**	**65.6**	**73.6**	**46.6**	**54.1**	**36.0**	**44.9**	**35.8**	**44.6**	**14.5**	**23.7**	**27.9**	**36.1**	**38.6**	**47.2**
Qwen2.5-7B Instruct																
CoT	3.5	11.0	18.7	25.4	8.8	14.1	7.2	15.9	6.7	15.2	2.8	8.3	13.8	19.6	8.8	15.6
RAG	30.6	37.2	58.0	64.2	38.1	45.4	26.4	34.0	24.9	33.3	7.4	14.7	16.3	24.3	28.8	36.2
Search-R1	40.6	48.0	65.2	73.7	39.7	49.2	38.4	45.6	38.9	45.6	14.7	21.4	37.1	44.0	39.2	46.8
- *pass@2*	41.5	-	67.4	-	50.6	-	40.2	-	40.6	-	15.6	-	45.9	-	43.1	-
ReSearch	41.6	49.8	64.0	71.6	42.8	50.8	39.2	46.0	40.8	47.2	15.8	25.3	43.1	49.7	41.0	48.6
WebSeer	40.1	48.4	65.5	73.2	42.3	48.7	38.6	45.6	**41.7**	**51.3**	14.4	24.6	42.6	48.9	40.7	48.7
ARR	**45.1**	**53.7**	**68.2**	**75.5**	**50.8**	**56.6**	**45.5**	**54.2**	41.5	50.6	**17.7**	**27.3**	**46.4**	**53.8**	**45.0**	**53.1**
Qwen3-8B																
CoT	2.7	10.7	19.0	25.7	8.5	14.7	7.4	16.7	7.8	16.9	5.3	13.6	17.4	26.3	9.7	17.8
RAG	34.1	42.7	58.5	67.2	40.5	47.2	31.4	36.2	30.7	35.4	10.6	17.4	23.1	31.7	32.7	39.7
Search-R1	42.6	50.6	67.5	75.1	42.2	48.5	41.8	47.3	42.3	48.6	16.3	24.9	42.6	53.0	42.2	49.7
- *pass@2*	42.9	-	70.5	-	48.2	-	45.4	-	48.8	-	19.6	-	48.9	-	46.3	-
ReSearch	38.5	46.7	62.3	70.8	40.7	49.7	38.1	47.4	39.2	45.1	15.2	23.1	40.6	49.7	39.2	47.5
WebSeer	46.3	52.8	68.0	75.6	41.7	50.4	46.2	53.4	47.0	54.8	14.6	23.0	44.9	55.6	44.1	52.2
ARR	**47.2**	**54.0**	**76.0**	**83.7**	**49.1**	**56.2**	**50.6**	**57.8**	**52.4**	**58.3**	**20.2**	**28.1**	**53.4**	**64.7**	**49.8**	**57.5**

Table 1: Performance comparison between ARR and baselines. Best and runner-up results are highlighted in **bold** and underline.

Variants	NQ	TQA	PQA	HQA	2Wiki	MSQ	BAM
Qwen2.5-3B Instruct							
ARR	53.2	73.6	54.1	44.9	44.6	23.7	36.1
w/o adv-out	50.6	69.6	51.5	43.2	42.9	22.4	35.5
w/o proc-adv	51.8	69.0	50.7	43.6	41.7	20.6	33.4
Qwen2.5-7B Instruct							
ARR	53.7	75.5	56.6	54.2	50.6	27.3	53.8
w/o adv-out	49.5	73.6	55.2	50.4	49.7	25.3	52.6
w/o proc-adv	49.6	72.5	51.6	45.2	48.3	21.9	47.0
Qwen3-8B							
ARR	54.0	83.7	56.2	57.8	58.3	28.1	64.7
w/o adv-out	53.7	83.4	55.6	57.3	56.1	27.5	63.2
w/o proc-adv	52.4	82.5	52.5	54.6	54.4	23.9	61.8

Table 2: Ablation studies on ARR. Datasets are abbreviated and correspond to Table 1, respectively.

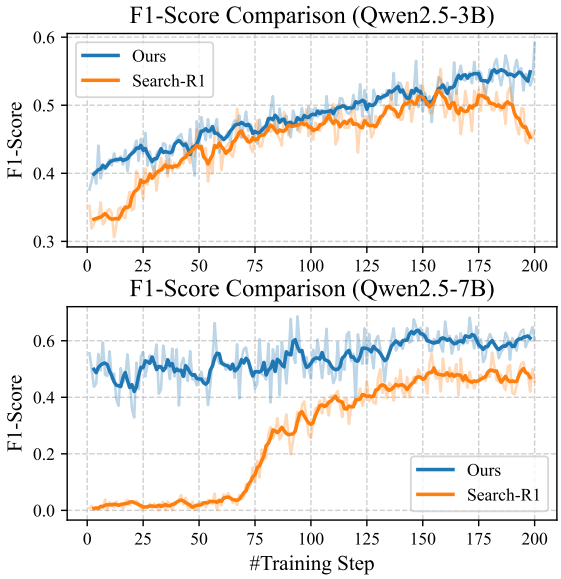

Figure 5: F1-score comparison.

2025a), ReSearch (Chen et al., 2025a), and WebSeer (He et al., 2025). Additionally, to fairly evaluate the efficacy of the adversarial yet cooperative dialogue in our multi-agent system, we introduce the **pass@2** metric for Search-R1.

Implementation For retrieval, all baselines adopt the same retriever and corpus setting as Search-R1. The retriever returns the top-3 documents. We select Qwen2.5-3B, -7B (Qwen et al., 2025), and Qwen3-8B (Yang et al., 2025) as backbone models. We optimize the policy model using the GRPO algorithm. For each prompt, we sample 5 trajectories with up to 5 interaction turns.

5.2 Main Results

Main results on general QA and multi-hop QA benchmarks across three backbone models are presented in Table 1. Overall, ARR consistently

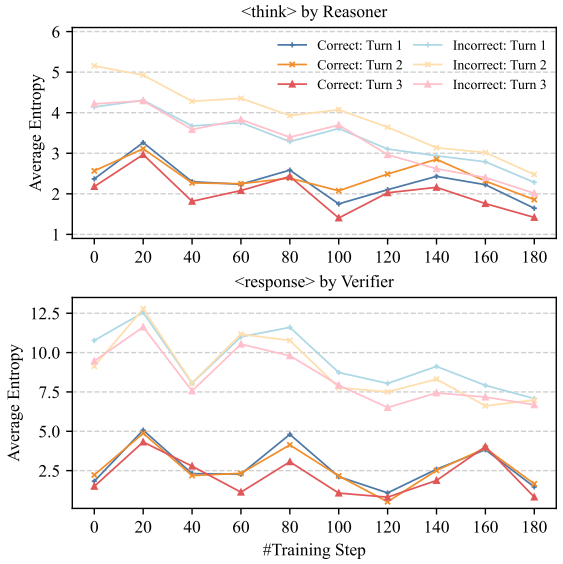

Figure 6: Entropy transition of agent actions in ARR.

outperforms all baseline methods across varying model sizes and datasets. The average improvement over runner-up baseline is 11.1% in EM and 7.6% in F1-score on Qwen2.5-3B, 9.5% in EM and 7.8% in F1-score on Qwen2.5-7B, and 13.4% in EM and 9.8% in F1-score on Qwen3-8B. The performance gains of ARR remain consistent as the model size scales from 3B to 8B, suggesting that the proposed method is model-agnostic.

Remarkably, ARR with 3B backbone outperforms baselines models with 7B backbone on general QA benchmarks. This indicates that multi-perspective reasoning unleash the potential of compact backbones on relatively simple benchmarks. ARR also exhibits significant performance gains on several multi-hop QA benchmarks. For instance, the gains over runner-up on Musique is 26.1%, 12.0%, and 23.9% in EM, respectively. Similarly, on HotpotQA, our method achieves the EM score of 0.455 and 0.506 on 7B and 8B. This shows that the multi-perspective reasoning architecture effectively solves complex multi-hop queries.

Our proposed method also frequently surpasses Search-R1 (**pass@2**). Take performance on NQ and HotpotQA with Qwen3-8B backbone for an example, ARR achieves 0.472 and 0.506 in EM, surpassing Search-R1 by 10% and 11.5%, respectively. This confirms that the superior performance of ARR is not the result of naive model scaling, but the adversarial yet cooperative dialogue and the multi-perspective optimization strategy.

Figure 5 shows the F1-score of our method and Search-R1 throughout training. Our method consistently outperforms Search-R1. Unlike Search-R1 which suffers from the cold start problem, our methods shows strong performance during early training stage on the 7B model.

5.3 Ablation Studies

In this sub-section, we present the results of ablation experiments to evaluate the contribution of key components in ARR. We introduce 2 variants: (1) ARR without adversarial outcome rewards (*w/o adv-out*) and (2) ARR without process-aware advantage (*w/o proc-adv*). Results across three backbone models are shown in Table 2.

The removal of the process-aware advantage leads to the most significant performance drop, particularly on multi-hop QA benchmarks. For instance, on Musique dataset with Qwen2.5-7B backbone, the F1 score drops from 27.3 to 21.9. This suggests that the proposed process-aware advantage is crucial for complex tasks requiring multi-step deduction. The exclusion of the adversarial outcome reward also results in a consistent performance degradation, and the impact is smaller.

5.4 Entropy Evolution

We present the entropy transition of agent actions in multi-turn trajectories of ARR in Figure 6. In general, the action entropy of the third turn consistently achieving lower values than initial turns. The entropy of `response` by Verifier shows dramatic differences between correct and incorrect trajectories. These observations are consistent with the empirical studies regarding entropy pattern in Section 3.2. The uncertainty within `think` by Reasoner gradually decreases as training progresses, indicating that the Reasoner is acquiring multi-turn reasoning capabilities.

6 Conclusion

In this paper, we introduced ARR, a multi-perspective agentic RAG framework that decouples reasoning and verification into an adversarial yet co-evolving system. Further, we bridged the gap between outcome-oriented reward and process-aware guidance by proposing an adversarial outcome reward and a process-aware advantage that reward the verifier for evidence-grounded, and uncertainty reducing feedback. Results show that our methods consistently outperform existing baselines and frequently exceed the pass@2 results of competitors.

Limitations

To evaluate our method and baselines using simple metrics like EM and F1-score, our paper primarily focuses on QA benchmarks where answers are concise and in short format. This means that our method may not generalize well on tasks that require long responses. Additionally, our method train the Reasoner and the Verifier on the same node using VeRL (Sheng et al., 2024) and vLLM (Kwon et al., 2023). Since vLLM (v0.10.1.1) does not allow offloading two rollout engine instance to the same GPU, we have to train the two model on separate group of GPUs. For instance, on a node with 8 GPUs, the reasoner and the verifier are loaded to 4 non-overlapping GPUs. This may lead to suboptimal GPU utilization efficiency.

Ethics Statement

All data and software utilized in our work are publicly available. We use the datasets from FlashRAG (Jin et al., 2025b), which use the MIT License. All datasets used for experiments permit use for academic research. Furthermore, we leverage several open-source frameworks and baselines. The vLLM and VeRL are governed by the Apache 2.0 License. All baseline models are accessed via their public repositories. Regarding artifact documentations, our work focuses on English. Datasets we use contain general world knowledge. No offensive or sensitive information is included.

Our proposed method is designed to promote the performance of LLMs on knowledge intensive tasks. It may benefit fields like education and scientific research. Though our work improves agentic reasoning RAG and multi-agent optimization, there exists the risk of generating inaccurate information.

References

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025a. Research: Learning to reason with search for llms via reinforcement learning.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. 2025b. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, Yang Qin, Yuan Wang, Quanxing Zha, Sunhao Dai, and Changhua Meng. 2025. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms.

Daocheng Fu, Jianbiao Mei, Licheng Wen, Xuemeng Yang, Cheng Yang, Rong Wu, Tao Hu, Siqi Li, Yufan Shen, Xinyu Cai, Pinlong Cai, Botian Shi, Yong Liu, and Yu Qiao. 2025. Re-searcher: Robust agentic search with goal-oriented planning and self-reflection.

Guanzhong He, Zhen Yang, Jinxin Liu, Bin Xu, Lei Hou, and Juanzi Li. 2025. Webseer: Training deeper search agents through reinforcement learning with self-reflection.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 737–740, New York, NY, USA. Association for Computing Machinery.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti,

Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025a. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025c. Webthinker: Empowering large reasoning models with deep research capability.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning.

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. 2024. Text embeddings by weakly-supervised contrastive pre-training.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Ting Wu, Xuefeng Li, and Pengfei Liu. 2025. Progress or regress? self-improvement reversal in post-training. In *The Thirteenth International Conference on Learning Representations*.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. 2025. Understanding the dark side of LLMs' intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27066–27101, Vienna, Austria. Association for Computational Linguistics.

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

A Appendix

A.1 Additional Preliminary Studies

Due to the limited space, we present analysis of policy entropy pattern in Search-R1 trajectories on Qwen2.5-7B in this subsection. From Figure 7, empirical studies on Qwen2.5-7B show similar pattern with studies on the 3B models. There is a positive correlation between correctness and decreasing entropy pattern. Similar to observations in Section 3.2, there exists a rise in the proportion of samples exhibiting an overall reduction in policy entropy. Quantitatively, for the Qwen2.5-7B backbone, the proportion rises from from 51.65% in the early phase to 57.96% in the late training phase.

A.2 Implementation Details

We use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge database and use E5 (Wang et al., 2024) as the retriever model. Our experiments are conducted on 8×A100 GPUs, with full parameter optimization and gradient checkpointing. We build our method based on VeRL (Sheng et al., 2024) and use vLLM (Kwon et al., 2023) to accelerate agent rollouts. It should be noted that AI assistants were utilized to aid in both the refinement of this manuscript and the implementation of the code.

A.3 Prompt Templates

Prompt for the Reasoner

To answer the given question, you will act as a reasoner working collaboratively with a retriever. Follow these steps carefully:

1. Reasoning Phase: When you receive a question, begin by reasoning about it inside <think> and </think>. This is where you analyze the problem and determine what you already know.

2. Identify Knowledge Gaps: If, during your reasoning, you realize that you lack some necessary information, you can request external knowledge by calling a search engine. To do this, write your query inside <search> and </search>.

3. Receive Search Results: After submitting your query, the verifier will process it and provide you with the top search results along with its opinion. This information will be enclosed between <feedback> and </feedback>.

4. Verification Phase: Every time you receive new information, you must first verify its relevance and usefulness. Conduct this verification inside <verify> and </verify>.

5. Update Reasoning: Based on the verified information, perform another round of reasoning inside <think> and </think>. Repeat steps 2–4 as many times as needed until you have enough information to answer the question.

6. Provide the Answer: Once you determine that no further external knowledge is required, provide your final answer directly inside <answer> and </answer>. Make sure to verify and think before answer the question. Keep your answer concise without additional explanations. For example: <answer> Beijing </answer>.

Always adhere strictly to the specified XML-like tags and respond only with the required elements.

Question: [QUESTION]

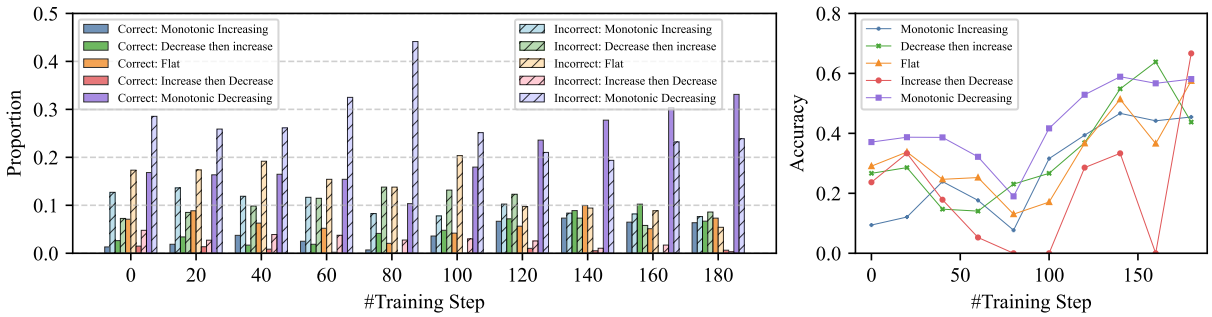

Figure 7: Statistical Analysis of Policy Entropy Pattern in Search-R1 trajectories. The y-axis of the **left subplots** denotes the proportion of trajectories exhibiting specific pattern relative to all multi-turn (≥ 3) samples. The y-axis of the **right subplots** represents the average accuracy of samples grouped by their pattens.

Prompt for the Verifier

As a verifier, your task is to collaborate with the reasoner to answer the given question. Follow these steps carefully:

1. Verification Process:
• The reasoner will provide its reasoning path, a retrieval query, and results from the search engine enclosed within <information> ... </information>.
• Perform a verification check inside <verify> ... </verify> to assess whether the query effectively contributes to answering the question.

2. Handling Effective Queries:
If the query is deemed appropriate:
• Choose the single most relevant document from the retrieved results and indicate it inside <selected_doc> ... </selected_doc> (e.g., <selected_doc> Doc 1 </selected_doc>).
• Synthesize the selected information and your own reasoning into a clear, concise reply inside <response>...</response>.

3. Handling Ineffective Queries:
If the query is judged ineffective, DIRECTLY Provide a justification for this assessment inside <response>...</response>.

4. Answer Verification:
If the reasoner provides an answer enclosed within <answer> and </answer>
• Verify the answer inside <verify> ... </verify> based on your judgment.
• Provide the final verified response inside <final_answer>...</final_answer>, ensuring it is concise and free of unnecessary details. For example: <final_answer>Beijing</final_answer>.
Always adhere strictly to the specified XML-like tags and respond only with the required elements.

Question: [QUESTION]

Prompt for the Final Predictor

The rollout text of the reasoner and verifier is: [REASONER & VERIFIER TRAJECTORY]

Answer the following question. Prior to this, both the reasoner and the verifier have conducted reasoning and verification regarding this question. You are required to provide the answer based on their respective reasoning processes. You should directly answer the question between without detailed illustrations.

Question: [QUESTION]