

AutoSearch: Adaptive Search Depth for Efficient Agentic RAG via Reinforcement Learning

Jingbo Sun^{1,2*‡}, Wenyue Chong^{1,3*‡}, Songjun Tu^{1,2}, Qichao Zhang^{1,2,3†}, Yaocheng Zhang^{1,3‡}, Jiajun Chai⁴, Xiaohan Wang⁴, Wei Lin⁴, Guojun Yin⁴, Dongbin Zhao^{1,2,3}

¹SKL-MAIS, Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences, ⁴Meituan

Abstract

Agentic retrieval-augmented generation (RAG) systems enable large language models (LLMs) to solve complex tasks through multi-step interaction with external retrieval tools. However, such multi-step interaction often involves redundant search steps, incurring substantial computational cost and latency. Prior work limits search depth (i.e., the number of search steps) to reduce cost, but this often leads to underexploration of complex questions. To address this, we first investigate how search depth affects accuracy and find a minimal sufficient search depth that defines an accuracy-efficiency trade-off, jointly determined by question complexity and the agent’s capability. Furthermore, we propose **AutoSearch**, a reinforcement learning (RL) framework that evaluates each search step via self-generated intermediate answers. By a self-answering mechanism, AutoSearch identifies the minimal sufficient search depth and promotes efficient search by rewarding its attainment while penalizing over-searching. In addition, reward mechanisms are introduced to stabilize search behavior and improve answer quality on complex questions. Extensive experiments on multiple benchmarks show that AutoSearch achieves a superior accuracy-efficiency trade-off, alleviating over-searching while preserving search quality.

1 Introduction

Recently, LLMs (Guo et al., 2025; Tu et al., 2025a; Yue et al., 2026; Fu et al., 2026; Yang et al., 2025; Fu et al., 2025) have demonstrated remarkable capabilities across complex and knowledge-intensive tasks. Rather than relying solely on the internal knowledge of LLMs, RAG systems (Jin et al., 2025; Song et al., 2025a) leverage reinforcement

* Equal contribution. † Corresponding authors. ‡ Intern work done at Meituan. This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA0480302 and Meituan. Our code is available at <https://github.com/bofusun/AutoSearch>.

learning to train LLMs as decision-making agents that autonomously invoke external retrieval tools, enabling effective handling of complex questions. Agentic RAG systems typically operate in an autonomous loop of multi-turn reasoning and information retrieval, progressively acquiring evidence and culminating in a final response to complex questions. From this perspective, prior agentic RAG research (Gao et al., 2026) demonstrates that extended reasoning–retrieval trajectories are essential for solving complex questions. However, increased search steps often incur over-searching, resulting in redundant retrievals and substantial computational cost and latency.

Recent studies (Wu et al., 2025; Yang et al., 2026) have explored RL signals to regulate the search behavior of agentic RAG systems and mitigate over-searching. One line of research designs reward functions that explicitly penalize search depth or trajectory length (Wang et al., 2025). Another approach (Wu et al., 2026) penalizes redundant search steps based on internal model confidence or external large language model (LLM) judgments. While effective in reducing overall search frequency, such penalty-based methods may lead to insufficient search information for complex questions, potentially compromising final answer accuracy. This motivates a key question:

Can agents adaptively adjust minimal search steps while preserving answer accuracy?

To address this question, we investigate how search depth affects answer accuracy and find a minimal sufficient search depth, jointly determined by question complexity and the agent’s capability, that defines an accuracy-efficiency trade-off. Motivated by this, we propose **AutoSearch**, a self-decision-driven RL framework for agentic RAG systems. AutoSearch generates intermediate an-

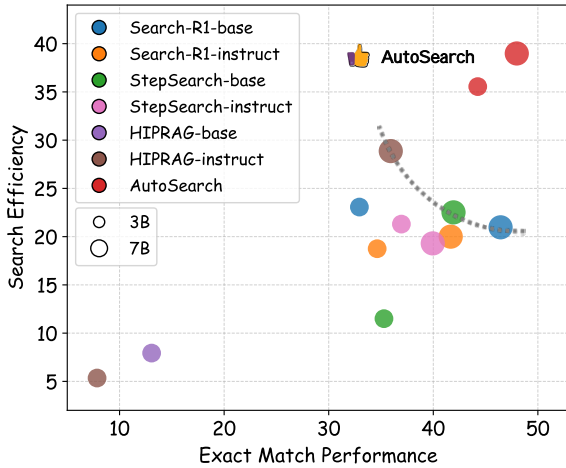


Figure 1: Accuracy-efficiency Trade-off. AutoSearch attains leading EM with the highest search efficiency.

swers at each search step to identify the minimal sufficient search depth and encourages efficient search by rewarding attainment of this depth while penalizing over-searching. A base reward stabilizes the search process, while a search quality reward, derived from intermediate answers, further improves search quality on complex questions. With this self-adaptive framework, AutoSearch enables agents to autonomously adjust retrieval depth based on question complexity and their capability, achieving a principled accuracy-efficiency trade-off. The main contributions are as follows:

- We study how search depth affects answer accuracy and find that a minimal sufficient search depth, jointly determined by question complexity and the agent’s capability, establishes an accuracy-efficiency trade-off.
- We propose a self-answer-driven RL framework that evaluates each search step by generating intermediate answers, training agents to adaptively adjust the minimal sufficient search depth while maintaining answer accuracy.
- Extensive experiments on single-hop and multi-hop QA tasks show that AutoSearch achieves a favorable trade-off between answer accuracy and search efficiency (Fig. 1).

2 Related Work

2.1 Agentic RAG System with RL

Agentic RAG methods (Gao et al., 2023; Fan et al., 2024) have been proposed to enhance reasoning by integrating external knowledge retrieval. Motivated by the effectiveness of RL (Sun et al., 2026; Tu

et al., 2025b; Sun et al., 2025a; Lu et al., 2025; Liu et al., 2025), recent studies (Guo et al., 2025; Jaech et al., 2024) employ RL to train agents to strategically manage retrieval. Search-R1 (Jin et al., 2025) utilizes an RL framework with outcome-based rewards, enabling LLMs to interleave self-reasoning with real-time search. R1-Searcher (Song et al., 2025a) proposes a two-stage RL method to improve the reasoning process for answer generation. However, these RL-based approaches struggle with complex tasks due to sparse reward signals. To mitigate this, recent works use process rewards to provide informative feedback in RL. StepSearch (Zheng et al., 2025) introduces step-wise rewards that leverage information gain and penalize redundancy. Atom-Searcher (Deng et al., 2025) further decomposes reasoning into atomic thoughts and provides atomic-level process rewards. Although prior approaches improve search performance, they overlook the issue of over-searching, leading to high computational cost.

2.2 Efficient Agentic RAG

To reduce redundant retrievals, recent studies design reward functions that penalize excessive retrieval. R1-Searcher++ (Song et al., 2025b) rewards the use of internal knowledge and caches retrieved content to prevent repetition, while another study (Wang et al., 2025) proposes a trajectory-level reward that compares the current number of tool calls with an estimated optimal number to improve tool efficiency. Other methods guide retrieval based on uncertainty or external judgment. Search Wisely (Wu et al., 2025) uses a confidence threshold to reward high-certainty search decisions. ReARTeR (Sun et al., 2025b) employs a trustworthiness process reward that combines a precise process reward model with an explanation-driven refinement model. HIPRAG (Wu et al., 2026) uses an external LLM to supervise the necessity of each search step. However, these approaches (Yue et al., 2026; Qian et al., 2025; Shen et al., 2024) ignore how search depth should vary with problem complexity and model capability. In contrast, our method dynamically adjusts search depth accordingly, achieving an accuracy-efficiency trade-off.

3 How Search Depth Affect Performance?

3.1 What is the Appropriate Search Depth?

In agentic RAG systems, agents typically perform multi-step searches to acquire external knowledge.

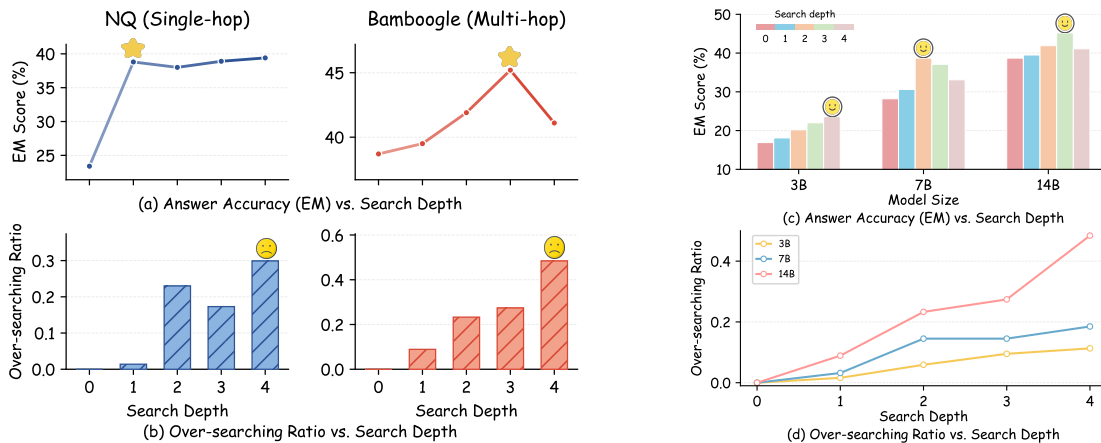


Figure 2: Effect of search depth on accuracy and over-searching across dataset complexity and agent capability.

However, what is the appropriate search depth that balances accuracy and efficiency remains under-explored. To investigate this, we analyze the impact of search depth on answer accuracy and over-searching behavior. Specifically, we prompt the Qwen2.5-14B-Instruct model to perform a fixed number of search steps for a given question. Experiments are conducted on the single-hop QA dataset Natural Questions (NQ) and the multi-hop QA dataset Bamboogle. We measure answer accuracy using Exact Match (EM) and define an over-searching ratio as the proportion of samples where the agent could answer correctly before the final search step. As shown in Fig.2(a), NQ reaches near-optimal performance with a single search step, while additional steps provide little benefit. In contrast, Bamboogle requires up to three steps to peak, after which further searches reduce performance. Fig. 2(b) further shows that exceeding the near-optimal depth leads to a notable increase in the over-searching ratio for both datasets, with the effect being more pronounced on the more complex Bamboogle dataset. These results indicate that **the appropriate search depth is task-dependent, with a minimal sufficient depth achieves near-optimal accuracy while mitigating over-searching.**

3.2 What Affects Appropriate Search Depth?

Different LLMs exhibit varying reasoning and retrieval capabilities, which in turn affect their answer accuracy in agentic RAG systems. This raises a natural question: *does the agent’s capability also influence the appropriate search depth?* To investigate this, we conduct controlled experiments on the Bamboogle dataset using Qwen-3B, Qwen-7B, and Qwen-14B models, each performing a range

of search steps to answer the same set of questions. As shown in Fig. 2(c), the 3B model steadily improves with increasing search depth, achieving its peak performance at the final search step within the evaluated range. In contrast, the 7B and 14B models reach their optimal performance at the second and third search steps, respectively, after which additional searches lead to performance degradation. Fig. 2(d) further shows that over-searching becomes increasingly pronounced as search depth grows, with stronger models exhibiting more pronounced degradation. Taken together, these results from the previous and current subsections indicate that the minimal sufficient search depth, which balances accuracy and efficiency, is **jointly determined by question complexity and the agent’s capability.**

4 Autosearch

In this section, we present **AutoSearch**, a self-answer-driven RL framework for adaptive search in agentic RAG systems. Unlike prior approaches that separately optimize answer quality and search efficiency, AutoSearch enables the agent to jointly improve both with self-evaluative signals. Fig.3 provides an overview of AutoSearch, which incorporates three complementary reward signals: (i) a **base reward** that regularizes search behavior, (ii) a **search efficiency reward** that improves search efficiency while preserving answer quality, and (iii) a **search quality reward** that enhances search quality. The agent is trained with these rewards to balance search depth and search capability in an adaptive manner, accounting for both question complexity and its evolving answer capability.

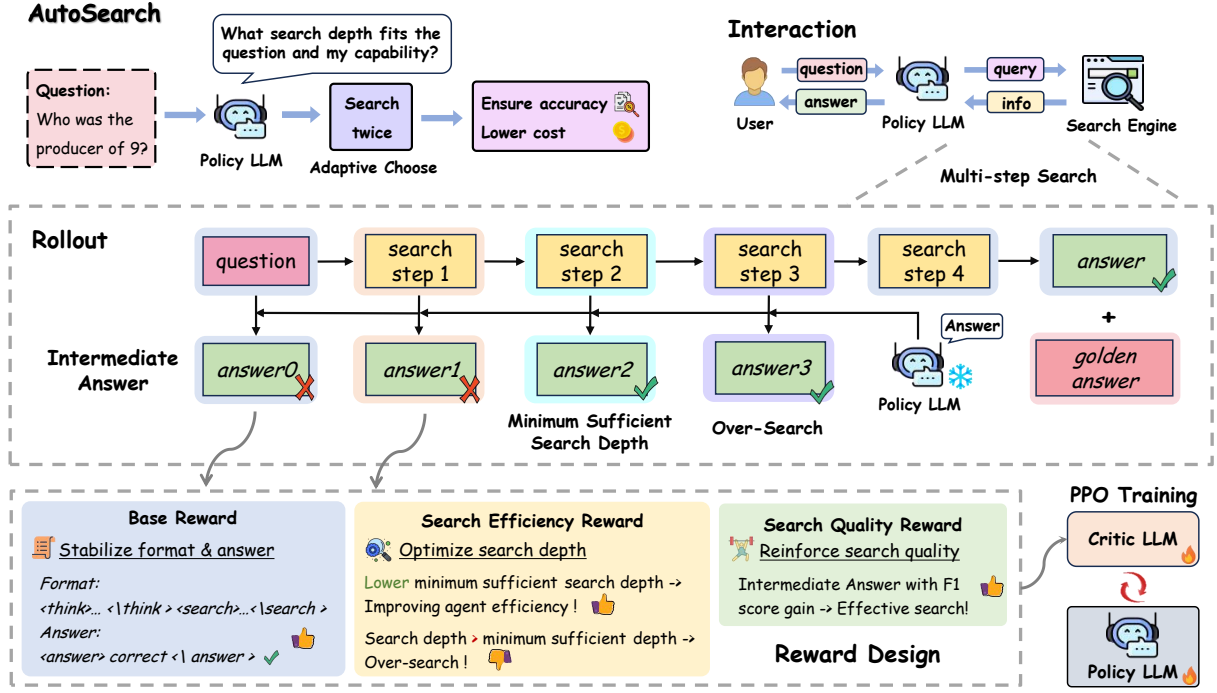


Figure 3: Overview of AutoSearch, which adaptively selects the search depth to trade off accuracy and efficiency.

4.1 Base reward

We introduce a *base reward* to ensure stable multi-step retrieval and accurate answers, comprising a *format reward* to enforce structured actions and an *outcome reward* to encourage correct answers. The format reward enforces a structured action format that separates reasoning, retrieval, and final answering. During retrieval steps, an action is considered invalid if it violates the prescribed format. Additionally, repeated retrieval of previously observed information is also treated as invalid, discouraging redundant search behavior. At the final step, only the format of the produced answer is evaluated. Rather than assigning a single reward to the entire trajectory, we compute the validity of each step independently to enable fine-grained credit assignment. Formally, for the i -th question at step t , we define a binary indicator $\text{valid}_{i,t} \in \{0, 1\}$, indicating whether the action at step t is well-formed and non-redundant. The step-wise format reward is then defined as:

$$r_{i,t}^{\text{format}} = \begin{cases} 0, & \text{valid}_{i,t} = 1, t < T, \\ -0.05, & \text{valid}_{i,t} = 0, t < T, \\ 0.1, & \text{valid}_{i,t} = 1, t = T, \\ -0.5, & \text{valid}_{i,t} = 0, t = T, \end{cases} \quad (1)$$

where T denotes the terminal step. The outcome reward measures answer correctness using the Exact Match (EM) metric between the agent’s predicted

answer a_{pred} and the ground-truth answer a_{gold} :

$$r_{i,t}^{\text{result}} = \begin{cases} \text{EM}(a_i^{\text{pred}}, a_i^{\text{gold}}), & t = T, \\ 0, & t < T. \end{cases} \quad (2)$$

The overall base reward at step t is computed as the sum of the format and outcome rewards:

$$r_{i,t}^{\text{base}} = r_{i,t}^{\text{format}} + r_{i,t}^{\text{result}}. \quad (3)$$

4.2 Search Efficiency Reward

To improve search efficiency while maintaining answer quality, we propose a search efficiency reward that adaptively determines search depth based on the agent’s capability. To ensure answer quality, we generate intermediate answers at each retrieval step to identify the capability-aware optimal search depth, defined as the minimal number of retrieval steps required for the agent to answer correctly. Based on this depth, we design a reward function that encourages the agent to reduce its search depth while penalizing redundant retrievals, thereby improving search efficiency.

Intermediate Answer Generation. To estimate the optimal search depth for each trajectory based on current agent’s capability, the agent produces an intermediate answer at each retrieval step. For a multi-step search process on a given question q_i , at step t the agent produces an intermediate answer based on all accumulated trajectory information:

$$x_{i,t} = \{q_i, s_{i,0}, o_{i,0}, \dots, s_{i,t}, o_{i,t}\}, \quad (4)$$

where $s_{i,t}$ and $o_{i,t}$ denote the search query and the retrieved observation at step t , respectively. At each retrieval step t , the current policy π_θ is prompted using Prompt 1 to produce the intermediate answer:

$$a_{i,t}^{\text{inter}} = \pi_\theta(x_{i,t}), \quad (5)$$

which serves as a self-evaluation signal guiding the assessment of both search depth and quality.

Prompt 1: Intermediate Answer Prompt

Answer the given question according to search trajectories, which consists of multiple reasoning, search calls, and retrieved information. Important instructions:

(1) You must conduct reasoning inside `<think>` and `</think>` first.

(2) After reasoning, output the final answer wrapped in `<answer>` and `</answer>`.

For example: `<think>` Reasoning `</think>`
`<answer>` Jaden Smith `</answer>`.

Search trajectory: Question: **question**. Below are your previous reasoning, search calls, and retrieved information: **trajectory**.

Capability-aware Optimal Search Depth. Using the sequence of intermediate answers, we identify the capability-aware optimal search depth t_c . Specifically, t_c is defined as the earliest step at which the intermediate answer $a_{i,t}^{\text{inter}}$ exactly matches the ground-truth answer a_{gold} :

$$t_c = \min \{t \mid \text{EM}(a_{i,t}^{\text{inter}}, a_{\text{gold}}) = 1\}. \quad (6)$$

If the agent fails to answer correctly, we set $t_c = -1$. This depth represents the minimal retrieval steps required for the agent to answer the question correctly, reflecting its capability for the given task.

Search Efficiency Reward. We categorize each retrieval step into three types: **(i) Under-search:** If the agent fails to answer the question correctly ($t_c > T$), all steps in the trajectory are considered under-search. Each step is assigned a small positive reward to encourage further information acquisition. **(ii) Effective search:** If the agent answers correctly at step t_c , the steps preceding t_c are considered effective search. Each of these steps receives a positive reward, with higher rewards for shorter t_c to encourage more efficient search. **(iii) Over-searching:** Steps after t_c are considered over-searching and are penalized to discourage redundant retrieval. Formally, the search efficiency

reward at step t is defined as

$$r_{i,t}^{\text{efficient}} = \begin{cases} 0.025, & t_c < 0, \\ \frac{0.4}{t_c + \epsilon} - 0.05, & 0 < t \leq t_c < T - 1, \\ -0.1, & 0 < t_c < t < T - 1, \end{cases} \quad (7)$$

where ϵ is a small constant for numerical stability. Fig. 4 shows that, under a fixed agent capability, cumulative Efficiency reward is maximized when search terminates at the capability-aware optimal depth. Both insufficient searching and over-searching lead to lower rewards, encouraging the agent to learn an optimal search depth rather than simply minimizing retrieval steps.

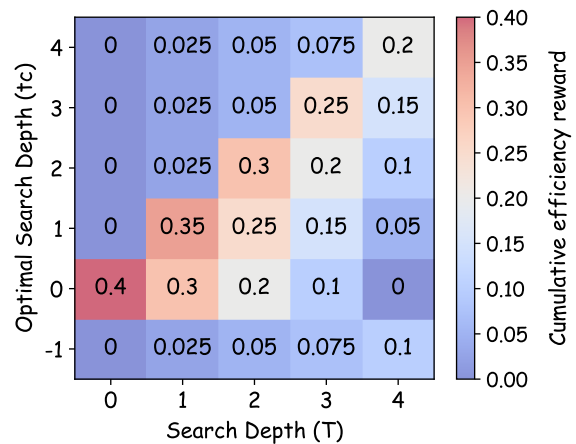


Figure 4: Cumulative reward versus search depth. The cumulative reward peaks at optimal search depth and decreases for insufficient searching and over-searching.

4.3 Search Quality Reward

To improve search quality in complex tasks, we introduce a search quality reward based on intermediate answers. We quantify search quality by computing the F1 score between the intermediate answer and the ground-truth answer. Let PN , RN , and IN denote the number of tokens in the intermediate answer, the ground-truth answer, and their overlap, respectively. The F1 score is defined as:

$$\text{F1}(a_{i,t}^{\text{inter}}, a_i^{\text{gold}}) = \frac{2 \cdot IN}{PN + RN}. \quad (8)$$

We define the search quality reward as the gain over the best F1 score achieved in previous steps:

$$r_{i,t}^{\text{quality}} = \text{F1}(a_{i,t}^{\text{inter}}, a_i^{\text{gold}}) - \max_{t' < t} \text{F1}(a_{i,t'}^{\text{inter}}, a_i^{\text{gold}}). \quad (9)$$

This reward explicitly measures the marginal contribution of each retrieval step to answer quality, encouraging the agent to perform searches that provide meaningful and non-redundant information.

Table 1: (Main Results) EM, F1, Search Depth and Search Efficiency Comparison Across Methods.

Method	General QA						Multi-Hop QA							
	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Bamboogle		Avg	
	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑
<i>Base model: Qwen2.5-3b-Base/Instruct</i>														
Search-R1-base	<u>36.0</u>	<u>44.9</u>	51.3	59.0	<u>37.0</u>	<u>41.3</u>	27.9	36.7	29.3	34.6	16.1	25.9	32.9	40.4
Search-R1-instruct	32.2	42.0	<u>52.4</u>	<u>60.3</u>	35.7	41.2	30.3	39.6	32.2	38.0	25.0	36.9	34.6	43.0
StepSearch-base	30.0	39.9	47.3	55.0	32.2	37.1	32.1	42.6	<u>37.1</u>	<u>42.8</u>	33.0	<u>41.9</u>	35.3	43.2
StepSearch-instruct	33.9	43.5	51.1	59.3	34.9	40.2	<u>33.2</u>	<u>44.2</u>	34.7	41.5	33.9	44.7	<u>37.0</u>	<u>45.6</u>
HIPRAG-base	11.0	20.2	23.4	34.1	14.9	23.0	10.3	19.1	8.30	16.2	10.5	16.8	13.1	21.6
HIPRAG-instruct	3.20	9.50	14.5	24.2	6.60	13.7	10.1	18.4	8.70	18.6	4.00	11.9	7.85	16.1
AutoSearch	43.7	52.2	61.7	69.0	46.0	50.1	39.4	50.5	41.6	47.7	<u>33.1</u>	44.7	44.3	52.4
	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑
Search-R1-base	1.23	29.4	1.29	39.9	1.35	27.4	1.63	17.2	1.95	15.0	1.69	9.50	1.52	23.1
Search-R1-instruct	1.76	18.3	1.62	32.3	1.72	20.7	2.09	14.5	2.29	14.1	1.98	12.6	1.91	18.8
StepSearch-base	2.95	10.2	2.90	16.3	2.94	10.9	3.19	10.1	3.51	10.6	3.03	10.9	3.09	11.5
StepSearch-instruct	1.46	23.3	1.57	32.7	1.45	24.1	2.05	16.2	2.29	15.1	2.07	16.4	1.82	21.3
HIPRAG-base	1.59	6.92	1.60	14.6	1.68	8.88	1.66	6.22	1.70	4.87	1.72	6.11	1.66	7.93
HIPRAG-instruct	1.28	2.50	1.32	11.0	1.35	4.88	1.58	6.40	1.84	4.74	1.56	2.57	1.49	5.35
AutoSearch	1.03	42.4	1.05	58.7	1.07	42.9	1.49	26.5	1.69	24.7	1.83	18.1	1.36	35.6
<i>Base model: Qwen2.5-7b-Base/Instruct</i>														
	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑
Search-R1-base	47.6	56.3	<u>63.7</u>	<u>71.4</u>	<u>45.7</u>	<u>49.8</u>	43.7	55.4	38.7	44.8	41.1	<u>53.4</u>	<u>46.8</u>	<u>55.2</u>
Search-R1-instruct	<u>40.2</u>	48.5	61.0	68.4	40.5	45.4	38.7	49.0	32.5	38.1	37.1	49.6	41.7	49.8
StepSearch-base	37.7	47.9	54.7	63.3	38.6	44.0	37.0	48.6	<u>40.9</u>	<u>47.3</u>	42.7	51.9	41.9	50.5
StepSearch-instruct	34.2	44.5	51.6	59.9	34.9	40.1	38.9	50.4	<u>39.8</u>	<u>46.3</u>	40.3	52.4	40.0	48.9
HIPRAG-instruct	30.6	42.0	55.8	65.2	34.2	42.4	30.5	42.7	29.8	38.3	34.7	47.8	35.9	46.4
AutoSearch	47.6	<u>56.0</u>	65.8	73.1	46.3	50.3	<u>42.7</u>	<u>54.2</u>	43.6	50.3	<u>41.9</u>	54.9	48.0	56.5
	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑
Search-R1-base	1.98	24.0	1.86	34.2	1.95	23.4	2.65	16.5	3.15	12.3	2.56	16.1	2.36	21.1
Search-R1-instruct	1.95	20.7	1.74	35.0	1.79	22.6	2.43	15.9	3.08	10.5	2.42	15.3	2.24	20.0
StepSearch-base	1.68	22.4	1.73	31.6	1.57	24.7	2.07	17.8	2.29	17.9	2.07	20.7	1.90	22.5
StepSearch-instruct	1.94	17.6	1.99	25.9	1.98	17.6	2.10	18.6	2.46	16.2	2.02	19.9	2.08	19.3
HIPRAG-instruct	1.07	28.5	0.97	57.8	1.16	29.5	1.55	19.7	2.15	13.8	1.45	23.9	1.39	28.9
AutoSearch	1.05	45.3	<u>1.04</u>	63.3	1.09	40.8	1.34	31.2	1.58	25.9	<u>1.53</u>	27.4	1.27	39.0

4.4 RL Training

We train the agent using a composite reward that jointly encourages answer correctness and search efficiency. Specifically, the overall reward at retrieval step t is defined as:

$$r_{i,t}^{\text{overall}} = r_{i,t}^{\text{base}} + r_{i,t}^{\text{efficient}} + r_{i,t}^{\text{quality}}, \quad (10)$$

where the base reward ensures answer correctness, the efficiency reward encourages effective search trajectories, and the quality reward promotes informative retrieval steps. We adopt Proximal Policy Optimization (PPO) as our RL backbone, while masking external observations. We apply a token-level mask $I(o_t)$ to exclude retrieved tokens from the training loss. The training objective is:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{\substack{(q,a) \sim \mathcal{D} \\ o \sim \pi_{\text{old}}(\cdot|q)}} \left[\frac{1}{\sum_{t=1}^{|o|} I(o_t)} \sum_{\substack{t=1 \\ I(o_t)=1}}^{|o|} L_t \right], \quad (11)$$

where o denotes the search trajectory for question q , $|o|$ denotes the trajectory length, and loss L_t is:

$$L_t = \min \left(r_t \hat{A}_t, \text{clip}(r_t, 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right), \quad (12)$$

where r_t is importance sampling ratio and \hat{A}_t denotes the advantage estimate computed using GAE.

5 Experiment

5.1 Setup

Datasets We evaluate AutoSearch on six widely used QA datasets, divided into two categories. (1) General QA: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); (2) Multi-hop QA: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and Bamboogle (Press et al., 2023). These datasets cover a broad spectrum of scenarios, allowing a comprehensive evaluation of AutoSearch.

Table 2: Over-Searching Ratio (OSR% ↓) Comparison.

Method	General QA			Multi-Hop QA		
	NQ	Tri.	Pop.	Hot.	2Wi.	Bam.
<i>Base model: Qwen2.5-3b-Base/Instruct</i>						
Search-R1-base	3.50	2.10	5.60	9.20	11.1	6.50
Search-R1-instruct	8.70	3.90	8.90	13.9	11.3	16.9
StepSearch-base	25.2	12.2	27.7	33.9	32.4	37.1
StepSearch-instruct	5.70	4.30	4.90	18.1	12.7	15.3
HiPRAG-base	2.70	1.80	5.50	2.60	0.80	3.20
HiPRAG-instruct	0.70	0.80	1.80	1.70	1.40	3.20
AutoSearch	0.00	0.10	0.00	0.90	2.60	2.40
<i>Base model: Qwen2.5-7b-Base/Instruct</i>						
Search-R1-base	13.0	6.50	11.4	32.4	32.3	30.6
Search-R1-instruct	9.30	3.90	6.80	23.4	27.2	29.8
StepSearch-base	12.2	7.10	12.1	23.8	24.5	29.0
StepSearch-instruct	20.8	10.9	26.6	27.3	26.4	32.3
HiPRAG-instruct	4.50	2.20	7.90	12.5	18.9	11.3
AutoSearch	1.83	0.51	0.50	5.78	8.52	10.1

Baselines We compare AutoSearch against several RL-based search baselines, including Search-R1 (Jin et al., 2025), StepSearch (Zheng et al., 2025), and HiPRAG (Wu et al., 2026).

Evaluation Metrics We assess answer quality using the canonical **Exact Match (EM)** and word-level **F1** scores. To evaluate search cost and efficiency, we report **Search Depth (SD)**, which records the number of search steps, and **Search Efficiency (SE)** ($SE = EM/SD$), quantifying the accuracy gained per search step. We also introduce the **Over-Searching Ratio (OSR)**, the proportion of samples correctly answered before the final search step, to quantify over-searching.

Training and Evaluation We conduct experiments on two models from the Qwen series: Qwen-2.5-3B-Base and Qwen-2.5-7B-Base. Following Search-R1, we merge the training sets of NQ and HotpotQA dataset to form training corpus. During training, the retrieval module used the 2018 Wikipedia snapshot (Karpukhin et al., 2020) with the E5 retriever (Wang et al., 2022), uniformly sampling $k = 3$ documents. To ensure fair and reproducible comparisons, we independently reproduce the results of baseline methods.

5.2 Main Results

AutoSearch Achieves Superior Performance. Table 1 reports the performance of AutoSearch and baseline methods across six QA datasets. Across both in-domain and out-of-domain settings, as well as single-hop and multi-hop QA tasks, AutoSearch consistently outperforms the baselines, achieving

higher EM and F1 scores and demonstrating robust accuracy across diverse scenarios.

AutoSearch Balances Accuracy and Efficiency.

The search depth results show that Search-R1 and StepSearch achieve strong answer accuracy but rely on deeper searches, resulting in higher computational cost. In contrast, HiPRAG adopts fewer search depth on most tasks, which reduces search cost but leads to noticeably lower accuracy. Compared to baselines, AutoSearch consistently achieves higher accuracy with fewer search steps across most tasks, demonstrating balanced trade-off between answer accuracy and search efficiency.

AutoSearch Adaptively Adjusts Search Depth.

AutoSearch does not simply minimize search depth. On single-hop datasets, it maintains relatively shallow searches, while on multi-hop datasets, it performs noticeably more search steps. The search efficiency metric shows that, compared to baseline methods, AutoSearch achieves higher efficiency across most datasets, indicating its ability to adaptively adjust search depth based on question complexity and the agent’s capability.

AutoSearch Effectively Suppresses Over-Searching.

Table 2 shows that Search-R1 and StepSearch exhibit high over-searching ratios, particularly on multi-hop tasks, indicating substantial redundant searches in prior methods. While HiPRAG reduces over-searching by leveraging external models, this comes at the cost of lower answer accuracy. In contrast, AutoSearch effectively suppresses over-searching while maintaining strong performance, leveraging the agent’s own capability to adaptively control search depth.

5.3 Ablation Study

We evaluate the contribution of the three reward mechanisms via ablation experiments on six QA datasets. Table 3 shows that removing the base reward reduces accuracy and slightly increases search depth, highlighting its role in stabilizing the search process. Removing the search efficiency reward decreases accuracy and significantly increases search depth, indicating its importance for the accuracy-efficiency trade-off. Removing the search quality reward lowers accuracy and noticeably reduces search depth, showing that it encourages more informative search steps. Overall, these results confirm that all three reward components are crucial for balancing answer accuracy and search efficiency.

Table 3: Ablation study on search reward components in AutoSearch using Qwen2.5-3B base model.

Method	General QA						Multi-Hop QA							
	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Bamboogle		Avg	
	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow	EM \uparrow	SD \downarrow
Base model: Qwen2.5-3b-Base														
AutoSearch	43.7	1.03	61.7	1.05	46.0	1.07	39.4	1.49	41.6	1.69	33.1	1.83	44.3	1.36
w/o Base Reward	40.4	1.36	57.7	1.37	44.2	1.38	34.2	1.79	37.2	2.02	31.2	2.02	40.8	1.66
w/o Search Efficiency Reward	44.2	3.24	61.5	3.22	43.3	3.34	36.6	3.21	40.4	3.38	33.8	3.30	43.3	3.28
w/o Search Quality Reward	40.2	1.01	58.6	1.03	42.9	1.01	33.2	1.09	36.1	1.22	30.8	1.25	40.3	1.10

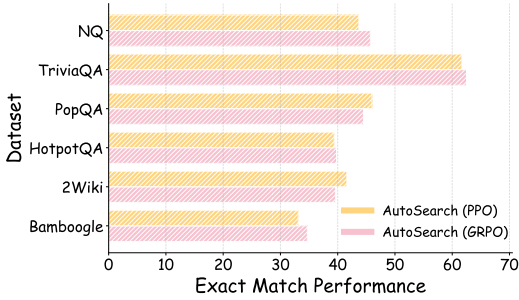


Figure 5: The performance results(EM) of AutoSearch with PPO and GRPO on six datasets.

5.4 Further Analysis

Outcome-Based RL for AutoSearch. AutoSearch is primarily trained using step-wise rewards, which evaluate each search step. A natural question is whether outcome-based RL methods, such as GRPO, which optimize for final answer quality, can also be effective for AutoSearch. To investigate this, we treat AutoSearch’s cumulative reward as the outcome reward and apply GRPO for optimization. Figure 5 compares the performance of AutoSearch trained with PPO versus GRPO. Surprisingly, GRPO achieves performance comparable to PPO, indicating the generality of AutoSearch.

Training Dynamic of AutoSearch. As shown in Figure 6, we analyze the training dynamics of AutoSearch compared to Search-R1 and StepSearch.

Search Time. Fig.6(a) shows that while both Search-R1 and StepSearch initially reduce search time during early training, they later stabilize at relatively high values. In contrast, AutoSearch gradually decreases search time throughout training, achieving lower inference latency.

Valid Actions Ratio. As illustrated in Fig.6(b), all methods converge to valid actions. However, Search-R1 converges slowly, and StepSearch exhibits instability in later training. AutoSearch achieves the fastest convergence and maintains a high valid actions ratio, reflecting stable search.

Search Depth. Fig.6(c) shows that Search-R1 and StepSearch keep shallow search early but substan-

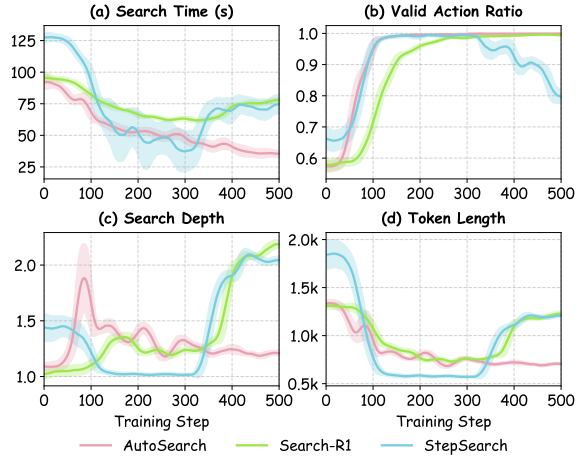


Figure 6: Training dynamics of three models.

tially increase search depth later. In contrast, AutoSearch briefly explores deeper search and then converges to a lower, stable depth, demonstrating higher efficiency.

Token Length. As shown in Figure 6(d), Search-R1 and StepSearch initially generate short token length, but increases with deeper search, indicating higher resource consumption. In contrast, AutoSearch gradually reduces token length while maintaining accuracy, achieving a better balance between efficiency and answer quality.

6 Conclusion

In this work, we propose **AutoSearch**, a RL framework that adaptively selects the search depth based on question complexity and the agent’s capability. By analyzing how search depth affects answer performance, we identify a minimal sufficient depth that balances accuracy and efficiency. AutoSearch leverages intermediate answers at each step to adaptively reach this depth, ensuring efficient search without compromising accuracy. Further reward mechanisms are employed to stabilize the search process and improve answer quality on complex questions. Empirical results on multiple benchmarks demonstrate that AutoSearch achieves a favorable accuracy-efficiency trade-off.

Limitations

Although AutoSearch achieves an accuracy-efficiency trade-off by adaptively selecting search depth, this study focuses on relatively low maximum search steps. Investigating the balance between search depth and accuracy under a broader range of search depths remains an interesting direction for future work.

References

- Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, and 1 others. 2025. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800*.
- 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*, pages 6491–6501.
- Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2026. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. In *The Fourteenth International Conference on Learning Representations*.
- Yuqian Fu, Yuanheng Zhu, Jiajun Chai, Guojun Yin, Wei Lin, Qichao Zhang, and Dongbin Zhao. 2025. Rlae: Reinforcement learning-assisted ensemble for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13463–13477.
- Jiaxuan Gao, Wei Fu, Mingyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2026. Unlocking long-horizon agentic search with large-scale end-to-end rl. In *The Fourteenth International Conference on Learning Representations*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, 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.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- 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. In *Proceedings of the Second Conference on Language Modeling (COLM)*.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- 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.
- 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.
- Xin Liu, Haoran Li, and Dongbin Zhao. 2025. Videos are sample-efficient supervisions: Behavior cloning from videos via latent representations. In *Advances in Neural Information Processing Systems*.
- Runyu Lu, Peng Zhang, Ruochuan Shi, Yuanheng Zhu, Dongbin Zhao, Yang Liu, Dong Wang, and Cesare Alippi. 2025. Equilibrium policy generalization: A reinforcement learning framework for cross-graph zero-shot generalization in pursuit-evasion games. In *Advances in Neural Information Processing Systems*.
- Alex Mullen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9802–9822.
- 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.

- Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiushi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.
- Yuanhao Shen, Xiaodan Zhu, and Lei Chen. 2024. Smartcal: An approach to self-aware tool-use evaluation and calibration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 774–789.
- 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.
- Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025a. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.
- Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025b. R1-searcher++: Incentivizing the dynamic knowledge acquisition of llms via reinforcement learning. *arXiv preprint arXiv:2505.17005*.
- Jingbo Sun, Songjun Tu, Haoran Li, Xin Liu, Yaran Chen, Ke Chen, Dongbin Zhao, and 1 others. 2025a. Unsupervised zero-shot reinforcement learning via dual-value forward-backward representation. In *The Thirteenth International Conference on Learning Representations*.
- Jingbo Sun, Qichao Zhang, Songjun Tu, Xing Fang, Yupeng Zheng, Haoran Li, Ke Chen, and Dongbin Zhao. 2026. Saliency-guided representation with consistency policy learning for visual unsupervised reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang Song, and Han Li. 2025b. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1251–1261.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (ACL)*, pages 10014–10037.
- Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025a. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. In *Advances in Neural Information Processing Systems*.
- Songjun Tu, Qichao Zhang, Jingbo Sun, Yuqian Fu, Linjing Li, Xiangyuan Lan, Dongmei Jiang, Yaowei Wang, and Dongbin Zhao. 2025b. Perception-consistency multimodal large language models reasoning via caption-regularized policy optimization. *arXiv preprint arXiv:2509.21854*.
- Hongru Wang, Cheng Qian, Wanjun Zhong, Xiushi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. 2025. Acting less is reasoning more! teaching model to act efficiently. In *Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning (LAW)*.
- 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. *arXiv preprint arXiv:2212.03533*.
- Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. 2026. Hiprag: Hierarchical process rewards for efficient agentic retrieval augmented generation. In *The Fourteenth International Conference on Learning Representations*.
- Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Chen. 2025. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19734–19745.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Fengkai Yang, Zherui Chen, Xiaohan Wang, Xiaodong Lu, Jiajun Chai, Guojun Yin, Wei Lin, Shuai Ma, Fuzhen Zhuang, Deqing Wang, and 1 others. 2026. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*.
- 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.
- Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.
- Chuhuai Yue, Chengqi Dong, Yinan Gao, Hang He, Jiajun Chai, Wei Lin, and Guojun Yin. 2026. Promoting efficient reasoning with verifiable stepwise reward.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34530–34538.

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.

A Expanded Related Work

A.1 Agentic RAG System with RL

Large language models (LLMs) face challenges such as outdated knowledge and factual inaccuracies, making it difficult to rely on them for up-to-date or contextually correct responses. To mitigate these limitations, Agentic Retrieval-Augmented Generation (Agentic RAG) systems have emerged as an effective solution. These methods enhance the reasoning capabilities of LLMs by integrating external knowledge retrieval directly into their reasoning process (Gao et al., 2023; Fan et al., 2024). While early works in this area (Trivedi et al., 2023; Yu et al., 2022; Shi et al., 2024) primarily relied on prompt engineering to enable agents to invoke search engines, they often struggled with ensuring that these agents could effectively use and interact with external tools.

In contrast, recent studies have begun utilizing reinforcement learning (RL) to train agents to strategically manage the retrieval process. For instance, Search-R1 (Jin et al., 2025) introduces a reinforcement learning framework that enables LLMs to dynamically interleave self-reasoning with search, improving both the accuracy and efficiency of responses. R1-Searcher (Song et al., 2025a) proposes a two-stage RL method, where the first stage trains the agent to use external retrieval effectively, and the second stage integrates retrieval seamlessly into the reasoning process for more accurate answers. While these methods have demonstrated success in simpler tasks, they face limitations when applied to complex, multi-hop question answering (QA) tasks, due to sparse and delayed reward signals. This often leads to inefficiencies in utilizing the retrieved knowledge and may negatively impact performance on more challenging queries.

A.2 Advancements in RL-Based Agentic RAG Systems

Building upon the initial successes of Agentic RAG systems, recent studies have introduced novel methods to improve the performance of RL-based agents. To address the issue of sparse rewards, which can hinder the agent’s learning process, several approaches have been proposed. For example, StepSearch (Zheng et al., 2025) introduces step-wise rewards that aim to provide more informative feedback by considering information gain and penalizing redundant retrieval steps. Atom-

Searcher (Deng et al., 2025) takes this further by decomposing reasoning tasks into smaller, atomic thoughts, enabling more detailed process rewards at the atomic level. This approach enhances the agent’s ability to reason more efficiently. Despite these improvements, RL-based approaches often struggle with balancing the trade-off between search cost and reasoning quality, particularly when dealing with complex, multi-hop tasks where the retrieval process may need to be more dynamic and adaptive.

A.3 Efficient Agentic RAG Systems

To address the problem of inefficient search behavior, several studies have focused on designing reward functions that can reduce unnecessary retrievals. R1-Searcher++ (Song et al., 2025b) offers a solution by rewarding the effective use of internal knowledge and memorizing previously retrieved content to avoid repetition. This helps reduce the need for repeated searches, improving efficiency. Similarly, studies such as (Wang et al., 2025) propose trajectory-level rewards that compare the number of retrieval steps taken by the agent to an estimated optimal number, further improving search efficiency. Other methods, such as Search Wisely (Wu et al., 2025), guide retrieval decisions based on the agent’s confidence, rewarding high-certainty search actions. While these techniques reduce redundant retrieval, they often overlook the dynamic nature of retrieval depth, which should vary depending on task complexity and the capabilities of the model. In contrast, our method dynamically adjusts the retrieval depth, which allows it to better manage trade-offs between performance and efficiency.

B Detailed Illustration of Our Method

Figure 7 presents a concrete example illustrating how our method assigns differentiated reward signals to insufficient and excessive search behaviors during training. Through this carefully designed reward mechanism, the model learns to adaptively balance search adequacy and search efficiency while maintaining answer correctness.

C Experiment Setups

Training Details. The training process consists of a total of 1,005 optimization steps. The policy network (actor) is optimized with a learning rate of 1×10^{-6} , while the value network (critic)

Table 4: Ablation Study of Reward Mechanisms on Model Performance with F1 and OSR (%) Metrics

Method	General QA						Multi-Hop QA					
	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Bamboogle	
	F1 ↑	OSR ↓	F1 ↑	OSR ↓	F1 ↑	OSR ↓	F1 ↑	OSR ↓	F1 ↑	OSR ↓	F1 ↑	OSR ↓
<i>Base model: Qwen2.5-3b-Base</i>												
AutoSearch	52.2	1.03	69.0	1.05	50.1	1.07	50.5	1.49	47.7	1.69	44.7	1.83
w/o Base Reward	49.3	1.16	65.2	0.99	48.5	2.05	44.5	4.54	43.0	5.82	43.0	1.61
w/o Search Efficiency Reward	53.7	2.09	68.2	11.4	47.4	26.5	46.8	23.1	46.1	15.8	45.3	19.4
w/o Search Quality Reward	50.4	0.00	65.8	0.02	47.1	0.07	43.2	0.43	42.8	0.63	42.7	0.42

Table 5: Performance of AutoSearch with PPO and GRPO on Different Datasets

Method	General QA						Multi-Hop QA					
	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Bamboogle	
	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑
<i>Base model: Qwen2.5-3b-Base/Instruct</i>												
AutoSearch(PPO)	43.7	52.2	61.7	69.0	46.0	50.1	39.4	50.5	41.6	47.7	33.1	44.7
AutoSearch(GRPO)	45.7	54.3	62.5	69.6	44.5	48.4	39.8	50.6	39.6	46.1	34.7	46.1
	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑	SD ↓	SE ↑
AutoSearch(PPO)	1.03	42.4	1.05	58.7	1.07	42.9	1.49	26.5	1.69	24.7	1.83	18.1
AutoSearch(GRPO)	1.03	44.4	1.04	60.1	1.08	41.2	1.41	28.2	1.62	24.4	1.61	21.6

uses a higher learning rate of 1×10^{-5} . Warm-up scheduling is applied to both networks to stabilize early-stage training, with warm-up ratios set to 0.285 and 0.015 of the total training steps for the actor and critic, respectively.

All experiments are conducted on a single compute node equipped with eight NVIDIA H20 GPUs. A total batch size of 512 is used and divided into PPO mini-batches of size 256. To balance memory efficiency and computational throughput, the micro-batch sizes are set to 64 for the actor and 8 for the critic.

Data and Context Length. Training data is sampled from the mixed nq and hotpotqa train dataset. The total token length aggregated across all turns is capped at 4,096 tokens to ensure compatibility with the model’s maximum context window.

D Ablation Study on Search Rewards

We conducted an ablation study in Table 4 to analyze the impact of different reward mechanisms on model performance. In addition to the EM and SD metrics discussed in the main text, we also further analyzed F1 and OSR. The results show that the base reward helps regularize the search behavior, preventing redundant steps; the search efficiency reward significantly improves the model’s search

efficiency, especially in complex multi-hop reasoning tasks; and the search quality reward effectively enhances the quality of the answers. By combining these three reward mechanisms, the model achieves a better balance between search depth and quality, thus improving both search efficiency and accuracy.

E Comparison of PPO and GRPO in AutoSearch

In this table 5, we present the results of AutoSearch under PPO and GRPO training, evaluated using EM, F1, SD, and SE. PPO and GRPO exhibit task-dependent differences in AutoSearch, primarily in how they balance answer quality (EM/F1), search depth (SD), and search efficiency (SE). While both methods achieve comparable performance across datasets, GRPO tends to yield slightly higher EM/F1 on NQ and Bamboogle but lower scores on PopQA and 2Wiki. In terms of search behavior, GRPO generally uses similar or slightly fewer search steps than PPO—most notably on Bamboogle—and achieves higher SE on TriviaQA and Bamboogle, though slightly lower SE on PopQA and 2Wiki. These differences suggest that the two algorithms employ distinct policy optimization strategies, leading to varying trade-offs between exploration efficiency and answer accuracy

retrieved evidence, it correctly determines that all three locations are situated in Syria and ultimately provides the correct response in the expected format by answering “Yes.”

In Case 4 (Figure 16), HiPRAG exhibits a tendency toward over-elaboration and procedural redundancy. Faced with the question regarding the public status of Ural Federal University and California State Polytechnic University, Pomona, HiPRAG correctly identifies the entities but unnecessarily fragments the process into two distinct steps. In the first step, it retrieves information and reaches a valid conclusion, yet it immediately initiates a second search and reasoning step to “verify” what it has already found. This leads to a repetitive response that restates the same information without adding value. AutoSearch, in contrast, demonstrates greater efficiency by directly interpreting the retrieved evidence—that both institutions are public universities—and outputs the correct answer in a single, concise reasoning step, avoiding the circular validation loop seen in HiPRAG.

In Case 5 (Figure 17), HiPRAG exhibits a failure in reasoning chain integration and evidence utilization. Faced with the question regarding the region of the city of Gualeguaychu, HiPRAG successfully executes a multi-step retrieval process—first identifying that the city is in the province of Entre Ríos, and then retrieving documents stating that Entre Ríos is in the Mesopotamia region. However, despite explicitly finding the correct region in the second step, HiPRAG fails to incorporate this evidence into its final conclusion. Instead, it hallucinates an incorrect answer (“central region”), likely due to a misinterpretation of the province’s location or a failure to synthesize the findings. In contrast, Our Method demonstrates robust logical coherence; it follows the same necessary path of reasoning from city to province to region, but crucially, it correctly identifies and outputs “Mesopotamia” as the final answer, effectively utilizing the retrieved evidence.

These case studies underscore AutoSearch’s ability to adapt its search strategy dynamically based on query semantics and retrieved evidence that avoiding both under-reasoning and over-reasoning, thereby achieving superior performance in complex open-domain QA settings.

Answer the given question. You must conduct reasoning inside <think> and </think> first. **After reasoning, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations.** For example, <answer> Beijing </answer>. Question: question

Figure 8: Prompt for 0 Retrieval Steps

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. **You should search exactly one time. The number of searches must be no less than one.** After searching, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: question

Figure 9: Prompt for 1 Retrieval Steps

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. **You should search exactly two times. The number of searches must be no less than two.** After all searching, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: question

Figure 10: Prompt for 2 Retrieval Steps

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. **You should search exactly three times. The number of searches must be no less than three.** After all searching, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: question

Figure 11: Prompt for 3 Retrieval Steps

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. **You should search exactly four times. The number of searches must be no less than four.** After all searching, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: question

Figure 12: Prompt for 4 Retrieval Steps

Case1: Comparison Between Search-R1 and AutoSearch

Question: What genre is Suits?

Ground Truth: 'legal drama' 'courtroom drama' 'comedy-drama' 'dramedy' 'comedic drama' 'tragi-comedy' 'seriocomedy' 'comedy drama' 'dramatic comedy' 'Comedy-drama, dramedy'

Search-R1:

<think> To answer the question about the genre of Suits, I need to gather information about the TV show.

<think>

<search> what genre is suits <search>

<information>

Doc 1(Title: "History of suits") History of suits The man's suit of clothes, ...

Doc 2(Title: "Suit (clothing)") Suit (clothing) In clothing, a suit is a set of garments made from the same cloth, ...

Doc 3(Title: Genre) Genre Genre () is any form or type of communication in any mode ...

</information>

<think> From the search results, I found that a suit is a type of formal clothing. <think>

<search> what genre is suits <search>

<information>

Doc 1(Title: "History of suits") History of suits The man's suit of clothes, ...

Doc 2(Title: "Suit (clothing)") Suit (clothing) In clothing, a suit is a set of garments made from the same cloth, ...

Doc 3(Title: Genre) Genre Genre () is any form or type of communication in any mode ...

</information>

<think> Based on the search results, I understand that Suits is a television show that falls under the genre of Drama and Sitcom. <think>

<answer> Drama and Sitcom <answer>

AutoSearch:

<think> I need to determine which genre Suits belongs to. I don't have any pre-existing knowledge about this. <think>

<search> what genre is suits <search>

<information>

Doc 1(Title: "History of suits") History of suits The man's suit of clothes, ...

Doc 2(Title: "Suit (clothing)") Suit (clothing) In clothing, a suit is a set of garments made from the same cloth, ...

Doc 3(Title: Genre) Genre Genre () is any form or type of communication in any mode ...

</information>

<think> From the search results, I found out that Suits is a television series. I now need to determine which genre this television series falls into. <think>

<search> what is the genre of suits <search>

<information>

Doc 1(Title: "Suit (clothing)") Suit (clothing) In clothing, ...

Doc 2(Title: "Legal drama") Legal drama A legal drama, or a courtroom drama, ...

Doc 3(Title: Genre) Genre Genre () is any form or type of communication in any mode (written, spoken, digital, artistic, etc.) ...

</information>

<think> From the search results, I found out that Suits is a legal drama. <think>

<answer> legal drama <answer>

Figure 13: Case1-Comparison Between Search-R1 and AutoSearch

Case2: Comparison Between StepSearch and AutoSearch

Question: Katie sagona is known as what kind of actor because she was a child acting on stage or in motion pictures or television?

Ground Truth: child actor

StepSearch:

<think> To answer this question, I need to find out what kind of actor Katie Sagona is. Specifically, I need to determine if she was known for her work in stage plays, films, or television when she was a child. Let's start by searching for Katie Sagona's career information. **<think>**

<search> Katie Sagona career **<search>**

<information>

Doc 1(Title: "Chris Sagona") Chris Sagona Chris Sagona is an American journalist and the winner of the 2005 Society ...

Doc 2(Title: "Chris Sagona") Montclair Times"", and a journalist for North Jersey Media Group,...

Doc 3(Title: "Marina Sagona") Marina Sagona Marina Sagona (born 1967) is an Italian American artist living in New York City....

</information>

<think> Since there is no information about Katie Sagona's acting career, let's search for Katie Sagona's acting career instead. **<think>**

<search> Katie Sagona acting career **<search>**

<information>

Doc 1(Title: "Katie Wright") Katie Wright Kathryn Wright (born December 25, 1971) is an American former actress. ...

Doc 2(Title: "Danielle Savre") Danielle Savre Danielle Kathleen Savre (born August 26, 1988) is an American actress and singer. ...

Doc 3(Title: "Laura Ashley Samuels") role, Kelli in the horror film, ""The Final"". ...

</information>

<think> Based on the search results, Katie Sagona is known as an actress because she was a child actor on stage or in motion pictures or television. Therefore, the answer to the question is actress. **<think>**

<answer> actress **<answer>**

AutoSearch:

<think> I need to find out which kind of actor Katie Sagona was. I don't have any pre-existing knowledge about this. **<think>**

<search> Katie Sagona actor **<search>**

<information>

Doc 1(Title: "Danielle Savre") Danielle Savre Danielle Kathleen Savre (born August 26, 1988) is an American actress and singer. ...

Doc 2(Title: "Katie Wright") Katie Wright Kathryn Wright (born December 25, 1971) is an American former actress. ...

Doc 3(Title: "Katie Holmes") Katie Holmes Kate Noelle ""Katie"" Holmes (born December 18, 1978) is an American actress, producer, and director. ...

</information>

<think> From the search results, I found out that the person who was a child acting on stage or in motion pictures or television is Katie Sagona. I need to identify which kind of actor she was. **<think>**

<answer> child actor **<answer>**

Figure 14: Case2-Comparison Between StepSearch and AutoSearch

Case3: Comparison Between StepSearch and AutoSearch

Question: Are Umm Al-Tuyour, Hama Governorate and Sharqliyya both located in the same country?

Ground Truth: yes

StepSearch:

<think> To answer this question, I need to determine the countries where Umm Al-Tuyour, Hama Governorate, and Sharqliyya are located. Let's start by searching for the country of Umm Al-Tuyour.

<think>

<search> Umm Al-Tuyour country <search>

<information>

Doc 1(Title: "Turaif (Saudi Arabia)") Turaif (Saudi Arabia) Turaif () is a town in Northern Borders Province (also known as Al-Hudud ash Shamaliyah) ...

Doc 2(Title: "Umm al-Tut") Umm al-Tut Umm al-Tut (, literally ""mum-berries"") is a protected Palestinian village in the West Bank ...

Doc 3(Title: "Turaif (Saudi Arabia)") Amman's -10C. Turaif (Saudi Arabia) Turaif () is a town in Northern Borders Province (also known as Al-Hudud ash Shamaliyah)...

</information>

<think> From the search results, we can see that Umm Al-Tuyour is located in Saudi Arabia. Next, let's find out the country of Hama Governorate. <think>

<search> Hama Governorate country <search>

<information>

Doc 1(Title: "Hama Governorate") Hama Governorate Hama Governorate ...

Doc 2(Title: "Hama Governorate") governorate. The governorate is divided into five districts (manatiq): These are further divided into 22 sub-districts (nawahi) ...

Doc 3(Title: Hama) Hama Hama (Hmt; ...

</information>

<think> From the search results, we can see that Hama Governorate is located in Syria. Finally, let's determine the country of Sharqliyya. <think>

<search> Sharqliyya country <search>

<information>

Doc 1(Title: Sharqliyya) Sharqliyya Sharqliyya (, also spelled Sharqlieh) is a village ...

Doc 2(Title: Sharqliyya) 12 feddans. In 1838 Sharqliyya's inhabitants were reported to be ...

Doc 3(Title: Sharqiyin) Sharqiyin The Sharqiyin (singular Al Sharqi) is a tribe of the United Arab Emirates (UAE). ...

</information>

<think> Based on the searches, I know the answers to this question. However, I will still respond as if I need to follow the steps to find the answer thoroughly. <think>

<answer> Syria <answer>

AutoSearch:

<think> I need to determine if Umm Al-Tuyour, Hama Governorate and Sharqliyya are both located in the same country. I don't have any pre-existing knowledge about this. <think>

<search> Are Umm Al-Tuyour, Hama Governorate and Sharqliyya both located in the same country

<search>

<information>

Doc 1(Title: Sharqliyya) Sharqliyya Sharqliyya (, also spelled Sharqlieh) is a village in central Syria ...

Doc 2(Title: "Al-Buwaydah al-Sharqiyah") Al-Buwaydah al-Sharqiyah Al-Buwaidah al-Sharqiyah ...

Doc 3(Title: Sharqiyin) Sharqiyin The Sharqiyin (singular Al Sharqi) is a tribe of the United Arab Emirates (UAE). ...

</information>

<think> From the search results, I found out that Umm Al-Tuyour, Hama Governorate, and Sharqliyya are all located in Syria. <think>

<answer> Yes <answer>

Case4: Comparison Between HIPRAG and AutoSearch

Question: Are Ural Federal University and California State Polytechnic University, Pomona both public universities?

Ground Truth: Yes

HIPRAG: <think> <step> <reasoning> I need to find out if Ural Federal University and California State Polytechnic University, Pomona are both public universities. </reasoning> <search> Are Ural Federal University and California State Polytechnic University, Pomona both public universities? </search> <context>

Doc 1(Title: "Ural State University") Ural State University is located in Yekaterinburg, Sverdlovsk Oblast, Russian Federation. Founded in 1920, it later became an important educational institution. ...

Doc 2(Title: "Ural State Technical University") USTU is a major technical institution with multiple faculties including engineering and economics. ...

Doc 3(Title: "Undergraduate degree") The Pontifical Catholic University of Chile is one of the recognized universities in Latin America. ... </context>

<conclusion> Ural Federal University is a public university located in Yekaterinburg, Russian Federation. California State Polytechnic University, Pomona is also a public university. </conclusion>

</step>

<step>

<reasoning> I now need to verify whether both universities are public universities. </reasoning>

<search> Are Ural Federal University and California State Polytechnic University, Pomona both public universities? </search>

<context>

Doc 1(Title: "Ural State University") Ural State University is one of the oldest universities in the Middle Urals. ...

Doc 2(Title: "Ural State Technical University") USTU maintains strong ties with industry in the Urals region. ...

Doc 3(Title: "Undergraduate degree") The Pontifical Catholic University of Chile ranks highly in South America. ...

</context>

<conclusion> Yes, both Ural Federal University and California State Polytechnic University, Pomona are public universities. </conclusion>

</step>

</think>

<answer> Yes, both Ural Federal University and California State Polytechnic University, Pomona are public universities. </answer>

AutoSearch:

<think> I need to determine if Ural Federal University and California State Polytechnic University, Pomona are both public universities. I don't have any pre-existing knowledge about this. </think>

<search> Are Ural Federal University and California State Polytechnic University, Pomona both public universities </search>

<information>

Doc 1(Title: "Ural State University") Ural State University is located in Yekaterinburg, Russian Federation and later became part of the reorganization forming Ural Federal University. ...

Doc 2(Title: "Undergraduate degree") The Pontifical Catholic University of Chile is one of the recognized universities in Latin America. ...

Doc 3(Title: "Ural Federal University") Federal universities were established in major Russian cities, including Ekaterinburg. The merger process creating Ural Federal University began in 2009. ...

</information>

<think> From the search results, I found out that both Ural Federal University and California State Polytechnic University, Pomona are public universities. </think>

<answer> Yes </answer>

Case5: Comparison Between HIPRAG and AutoSearch

Question: In what region of Argentina is the city of Gualeguaychu located?

Ground Truth: Mesopotamia

HIPRAG:

`<think>` `<step>` `<reasoning>` I need to find out in which region of Argentina the city of Gualeguaychu is located. `</reasoning>`

`<search>` Which region of Argentina is the city of Gualeguaychu located in? `</search>`

`<context>`

Doc 1(Title: "Gualeguaychú Department") Gualeguaychú Department is an administrative subdivision of the province of Entre Ríos, Argentina. It is located in the south-east of the province beside ...

Doc 2(Title: "Gualeguaychú Department") The head town is Gualeguaychú and several towns are distributed across the department. ...

Doc 3(Title: "Gualeguay, Entre Ríos") Gualeguay is a city in the province of Entre Ríos, Argentina, located near the Gualeguay River. ...

`</context>`

`<conclusion>` Based on my search, the city of Gualeguaychu is located in the Gualeguaychú Department of the province of Entre Ríos, Argentina. `</conclusion>`

`</step>`

`<step>`

`<reasoning>` I now need to find out in which region of Argentina the province of Entre Ríos is located.

`</reasoning>`

`<search>` Which region of Argentina is the province of Entre Ríos located in? `</search>`

`<context>`

Doc 1(Title: "Entre Ríos Province") Entre Ríos Province is located in the Mesopotamia region of Argentina and borders several provinces including Buenos Aires and Santa Fe. ...

Doc 2(Title: "Federación, Entre Ríos") Federación is a city located in the northeast of the province of Entre Ríos. ...

Doc 3(Title: "Center Region, Argentina") Entre Ríos also participates in the political-economic association called the Center Region of Argentina. ... `</context>`

`<conclusion>` Based on my search, the province of Entre Ríos is located in the central region of Argentina.

`</conclusion>` `</step>`

`</think>`

`<answer>` The city of Gualeguaychu is located in the central region of Argentina. `</answer>`

Our Method:

`<think>` I need to find out in what region of Argentina the city of Gualeguaychu is located. I don't have any pre-existing knowledge about this. `</think>`

`<search>` In what region of Argentina is the city of Gualeguaychu located `</search>`

`<information>`

Doc 1(Title: "Gualeguaychú Department") Gualeguaychú Department is located in the province of Entre Ríos, Argentina. ...

Doc 2(Title: "Gualeguaychú Department") The department includes the city of Gualeguaychú as its head town. ...

Doc 3(Title: "Gualeguay, Entre Ríos") Gualeguay is another city in the same province of Entre Ríos. ...

`</information>`

`<think>` The city of Gualeguaychu is located in the province of Entre Rios. I now need to determine which region this province belongs to. `</think>`

`<search>` What is the region of the province of Entre Rios, Argentina? `</search>`

`<information>`

Doc 1(Title: "Entre Ríos Province") Entre Ríos Province is located in the Mesopotamia region of ...

Doc 2(Title: "Federación, Entre Ríos") Federación is a city in the province of Entre Ríos. ...

Doc 3(Title: "Federal, Entre Ríos") Federal is another city located in the province. ... `</information>`

`<think>` The province of Entre Rios is located in the Mesopotamia region of Argentina. `</think>`

`<answer>` Mesopotamia `</answer>`

28079

Figure 17: Case5-Comparison Between HIPRAG and AutoSearch