Careful Queries, Credible Results: Teaching RAG Models Advanced Web Search Tools with Reinforcement Learning

Yuqin Dai[1,5] *, Shuo Yang[2*], Guoqing Wang[5*], Yong Deng [5], Zhanwei Zhang[3,5], Jun Yin[1], Pengyu Zeng [1], Zhenzhe Ying [5], Changhua Meng [5], Can Yi [5], Yuchen Zhou [4], Weiqiang Wang [5], Shuai Lu [1]

[1]Tsinghua University, [2]The University of Hong Kong, [3]Zhejiang University,
[4]National University of Singapore, [5]Ant. Group

Abstract

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by integrating up-to-date external knowledge, yet real-world web environments present unique challenges. These limitations manifest as two key challenges: pervasive misinformation in the web environment, which introduces unreliable or misleading content that can degrade retrieval accuracy, and the underutilization of web tools, which, if effectively employed, could enhance query precision and help mitigate this noise, ultimately improving the retrieval results in RAG systems. To address these issues, we propose WebFilter, a novel RAG framework that generates source-restricted queries and filters out unreliable content. This approach combines a retrieval filtering mechanism with a behavior- and outcome-driven reward strategy, optimizing both query formulation and retrieval outcomes. Extensive experiments demonstrate that WebFilter improves answer quality and retrieval precision, outperforming existing RAG methods on both in-domain and out-of-domain benchmarks. Code is available at https://github.com/GuoqingWang1/WebFilter.

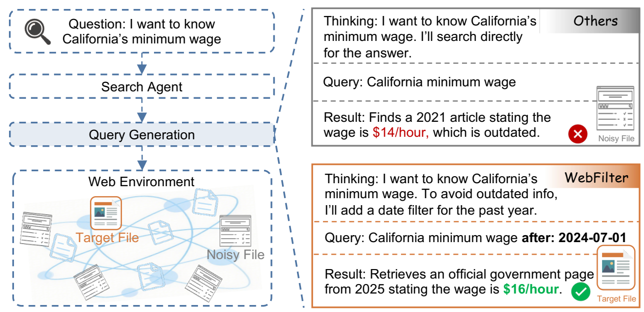


Figure 1: Comparison of WebFilter with Existing Methods: Existing methods (Zheng et al. 2025; Song et al. 2025) often yield unreliable results in misinformation-rich web environments. WebFilter enhances accuracy by using advanced search operators to filter noise and retrieve target files.

Introduction

The advancement of large language models (LLMs) has driven significant progress across both industrial and academic fields (Qiu et al. 2024; Zhang et al. 2024; Yu et al. 2025). Despite the wide applicability of LLMs, they often struggle with knowledge-intensive queries because their knowledge can be incomplete or outdated, which leads to factual inaccuracies or hallucinations (Zhang et al. 2023; Sahoo et al. 2024; Ji et al. 2023). To address these challenges, Retrieval-Augmented Generation (RAG) enhances model performance by retrieving relevant external knowledge during inference. This approach enables the model to access up-to-date information and fill in knowledge gaps.

RAG is meant to help models access up-to-date information, but the high cost and delay of online retrieval have caused early research (Chen et al. 2025; Jin et al. 2025; Song et al. 2025) to focus mainly on using locally stored knowledge. While these local sources are efficient, they are often outdated or incomplete, which affects model performance in real-world situations. More recent research (Li et al. 2025b; Zheng et al. 2025; Wei et al. 2025) has started exploring RAG in web-based environments, showing the benefits of using web search during model training. However, unlike local retrieval, which relies on trusted, static data, web-based retrieval comes with its own challenges: 1) **Web environment pervasive misinformation**: the open web is saturated with misinformation, low-quality content noise (Yang et al. 2025b,c). This significantly increases the difficulty of identifying credible sources and introduces risks of model hallucination or factual inconsistency during answer generation. 2) **Web tools underutilization**: local tools and web-based search engines differ significantly in utilization. While web tools provide *advanced search operators* that help avoid outdated information and enable retrieval from trusted sources, such capabilities are unavailable in offline settings. As a result, locally trained models struggle to learn and use advanced tools, limiting their ability to filter noise and focus on reliable sources.

To address these challenges, we present WebFilter, a framework that improves answer accuracy by filtering noise using advanced search operators. Given the imperfections of search engines and their sensitivity to query quality (see Fig. 1), our framework further enhances retrieval accuracy by formulating more effective queries and applying advanced operators. These operators, such as source selection and time filtering, enable precise retrieval by filtering out

*These authors contributed equally.

noisy sources and enhancing credibility. However, guiding models to correctly use these operators is challenging. Our experiments show that, even with instructions, models rarely proactively use advanced operators. This is because, while existing models (Li et al. 2025b; Zheng et al. 2025; Song et al. 2025) have achieved state-of-the-art results using Reinforcement Learning (RL) (Kaelbling, Littman, and Moore 1996), they focus primarily on outcomes, rather than guiding behavior in web-based environments. As a result, they often rely on unreliable shortcuts. Thus, methods that effectively guide models to utilize advanced operators for noise filtering are crucial.

Therefore, to systematically integrate web search into model reasoning, we formulate retrieval as a Markov Decision Process (MDP) and guide the model to operate as an information retrieval agent capable of using advanced operators. To more effectively guide model behavior, we introduce an Information-Filtering Reward strategy, which combines two complementary rewards driven by both behavioral and outcome considerations. Specifically: (1) **Source-restricting Reward (SR)** encourages the model to proactively use advanced search operators (e.g., domain filters, date ranges), shaping query formulation strategies. In the early stages of training, SR promotes exploration even when the model's performance is suboptimal. As training progresses, SR gradually shifts focus towards accuracy, refining the model's query formulation to prioritize precision. This transition helps balance exploration and exploitation, ensuring effective learning and improved retrieval performance. (2) **Retrieval-precision Reward (RR)** reinforces outcomes by having a more capable, large-scale LLM evaluate the quality of retrieved content and provide feedback, enabling the model to refine its queries and improve source selection based on retrieval results. Through the combination of structured modeling, instruction design, and reward learning, WebFilter overcomes pervasive web misinformation and fully leverages advanced web search tools for precise and reliable retrieval. In summary, our contributions are as follows:

- We propose WebFilter, a novel RAG framework explicitly designed for real-world web environments. It formulates retrieval as an MDP and trains LLMs as information retrieval agents, enabling effective mitigation of pervasive misinformation and better utilization of advanced web search tools.
- We introduce an information-filtering reward strategy that guides precise, source-restricted retrieval and enables robust misinformation filtering, addressing both pervasive web noise and tool underutilization.
- Experiments show that WebFilter achieves state-of-the-art QA performance, with advanced search operator usage rising from 10% to 75%, and significant gains across in-domain and out-of-domain benchmarks.

Related Work

Agentic Retrieval Augmented Generation

Recent work has explored agentic RAG (Chen et al. 2025; Jin et al. 2025; Song et al. 2025) to integrate retrieval into the reasoning process of LLMs. For example, methods such as ReSearch (Chen et al. 2025), Search-R1 (Jin et al. 2025), and R1-Searcher (Song et al. 2025) train LLMs to autonomously generate search queries while reasoning with a local search engine. However, LLMs trained in such local settings often struggle to generalize to real-world web environments (Zheng et al. 2025). To overcome this, methods such as WebRL (Qi et al. 2024), WebThinker (Li et al. 2025b), R1-Searcher (Song et al. 2025), DeepResearcher (Zheng et al. 2025), WebAgent-R1 (Wei et al. 2025) leverage online search engines for training. Yet, compared to local settings, online environments pose greater challenges, including high API costs, network latency, and the abundance of false or redundant information, all of which hinder efficient training and retrieval (Zheng et al. 2025). However, due to their reliance on local web environments and the lack of source-specified retrieval data, existing reward schemes fall short in tackling real-world web challenges such as pervasive misinformation and poor use of advanced search operators. To address this issue, we formulate retrieval as a Markov Decision Process, guided by explicit instruction on tool usage and an Information-Filtering Reward strategy, jointly enabling more structured, source-restricted querying and robust information filtering.

Reinforcement Learning for LLMs

Reinforcement learning (RL) has become increasingly prominent in training LLMs, supporting applications that span from preference alignment (Ouyang et al. 2022; Casper et al. 2023; Kaufmann et al. 2023) to complex tasks (Hao et al. 2023; Pang et al. 2024; Tang et al. 2025; Xie et al. 2025). A growing area of interest is the application of RL to tool-integrated tasks (Li, Zou, and Liu 2025), which involve multi-step interactions and dynamic tool states. The high interactivity with the environment makes them a natural fit for RL. Existing research has explored RL-trained LLM agents for tool-integrated reasoning. For example, ToRL (Li, Zou, and Liu 2025) and Tool-N1 (Zhang et al. 2025) employ rule-based outcome rewards that account for both accuracy and format to guide RL, while other methods (Wang et al. 2025; Sha, Cui, and Wang 2025; Singh et al. 2025) extend this by incorporating tool usage-based reward. However, most RL methods are built on local corpora without source-restricted supervision, limiting their generalization to real-world web environments with noisy information and underused search tools. We address this by formulating retrieval as an MDP and combining tool-use instruction with an Information-Filtering Reward strategy.

Methodology

In this section, we introduce the WebFilter training framework, designed to enhance Retrieval-Augmented Generation (RAG) by improving query formulation and filtering unreliable web content. As shown in Fig. 2, we model the retrieval process as a Markov Decision Process, enabling the model to decide when and how to issue search queries and integrate the retrieved information. To guide this process, we implement an Information-Filtering reward strategy, which evaluates retrieval outcomes and refines query strategies based on feedback from a stronger LLM. The following sections detail the framework and problem formulation.

Problem Formulation

We model the task completion as a Markov Decision Process (MDP), denoted by $(S, A, R, \mathcal{T})$, where the state $s_t \in S$ represents the history of previous actions at time step t. At each t step, the agent selects an action $a_t \in A$ based on the current state s_t, following the policy π_θ. When the agent selects the "search" action ($a_t =$ search), it updates the state by incorporating the retrieved results. Specifically, d_t refers to the content retrieved based on the search query at time step t. The state transition is defined as:

$$s_{t+1} = T(s_t, a_t) = \begin{cases} [s_t; a_t, d_t] & \text{if } a_t = \text{search}, \\ [s_t; a_t] & \text{otherwise.} \end{cases} \tag{1}$$

where T represents the deterministic state transition function, and the agent receives a reward $r_t = R(s_t, a_t)$, as determined by the environment. The process terminates when the task is completed or the maximum allowed interactions are reached.

Learning to Use Advanced Search Tools

WebFilter is implemented as a prompt-based Information Retrieval Agent that proactively conducts web searches and reasons over retrieved results before issuing a final answer. It operates under a cost-sensitive policy, minimizing excessive queries and avoiding uncertain responses when evidence is lacking. The agent is explicitly instructed to integrate advanced search operators such as `OR`, `AND`, `NOT`, quotation marks for exact phrases, domain restrictions via `site:`, and date filters like `after:`, enabling precise, source-restricted retrieval. It also prioritizes trusted domains (e.g., `wikipedia.org`, `gov.cn`), guiding the model to generate focused, reliable, and efficient queries. For implementation details, please refer to our GitHub repository.

Information-Filtering Reward Strategy

Although the MDP formulation structures retrieval interactions, it alone cannot ensure precise query formulation or reliable filtering of web content. To address this, we propose an Information-Filtering Reward strategy that integrates both behavior-based and outcome-based incentives. The Source-restricting Reward (SR) acts as a behavior-based restriction, promoting the use of advanced search operators for precise, source-restricted queries. In contrast, the Retrieval-precision Reward (RR) serves as an outcome-based signal, leveraging external critique to assess and refine retrieval quality. Together, these rewards guide the model toward more effective and trustworthy web search. Next, we will describe its design in detail.

Source-restricting Reward (SR) To encourage precise and source-restricted queries, we design a rule-based Source-restricting Reward that promotes the use of advanced search operators. Specifically, the Source-restricting Reward is defined by the following formula:

$$\mathcal{K} = \{k_1, k_2, ..., k_m\} \subseteq \Sigma^*, \tag{2}$$

$$\mathcal{Q} = \{q_1, q_2, ..., q_n\} \subseteq y, \tag{3}$$

$$R_{src} = \mathbb{I}\left[\exists q \in \mathcal{Q},\ \exists k \in \mathcal{K},\ ReMatch(k, q) = 1\right], \tag{4}$$

where Σ^* denotes the set of all sequences over the vocabulary of the policy model; $\mathcal{K} \subseteq \Sigma^*$ is a predefined set of advanced search keyword patterns, such as "site:", "-", or "AND"; $y \subseteq \Sigma^*$ is a response generated by the policy model; and $\mathcal{Q} \subseteq y$ is the set of search queries extracted from y. The binary function $ReMatch(k, q)$ returns 1 if query q matches the regular expression associated with pattern k, and 0 otherwise. The indicator function $\mathbb{I}[\cdot]$ equals 1 if the predicate holds and 0 otherwise. We define the Source-restricting Reward $R_{src} \in \{0, 1\}$ to be 1 if any query in $\mathcal{Q}$ contains an advanced search pattern from $\mathcal{K}$, and 0 otherwise. This design explicitly promotes the use of advanced search operators in the reinforcement learning process.

Retrieval-precision Reward (RR) Beyond encouraging the use of advanced search operators, it is equally important to ensure they are applied correctly and effectively. We thus propose an LLM-based, outcome-oriented Retrieval-precision Reward (RR) that evaluates responses and provides feedback on operator correctness and quality. Specifically, the Retrieval-precision Reward is defined as:

$$z,\, c = LLM_{Judge}(g, y, I_{Judge}), \tag{5}$$

where LLM_{Judge} denotes a powerful LLM used to evaluate the model's predictions, g is the ground-truth answer, y is the response generated by the policy model, and I_{Judge} denotes the prompt template provided to LLM_{Judge}, as shown in . The scalar $z \in \{0, 1\}$ measures the correctness of the predicted answer, while $c \in \{0, 1\}$ evaluates whether the use of advanced search syntax contributed to the retrieval quality.

Reward Aggregation The final reward R is computed as:

$$R = \begin{cases} -1 & \text{if } \neg C_{\text{format}}, \\ f(z, c, R_{f1}) & \text{if } C_{\text{format}} \wedge C_{\text{agg}}, \\ 0.1 & \text{if } C_{\text{format}} \wedge \neg C_{\text{agg}} \wedge C_{\text{src}}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Here, C_{format} denotes correct output format; C_{agg} indicates $f(R_{llm}, R_{f1}) \neq 0$; C_{src} means $R_{src} = 1$. The F1 Reward R_{f1} is computed as:

$$R_{f1} = \frac{2 \times IN}{PN + RN}, \tag{7}$$

where PN is the word count of the predicted answer, RN is the word count of the reference answer, and IN is the word count of overlapping words between them. The aggregation function $f(\cdot)$ is defined as:

$$f(u, v, w) = \alpha u + \beta v + (1 - \alpha - \beta) w, \tag{8}$$

where α and β are hyperparameters used to balance the influence of each reward on the policy optimization process.

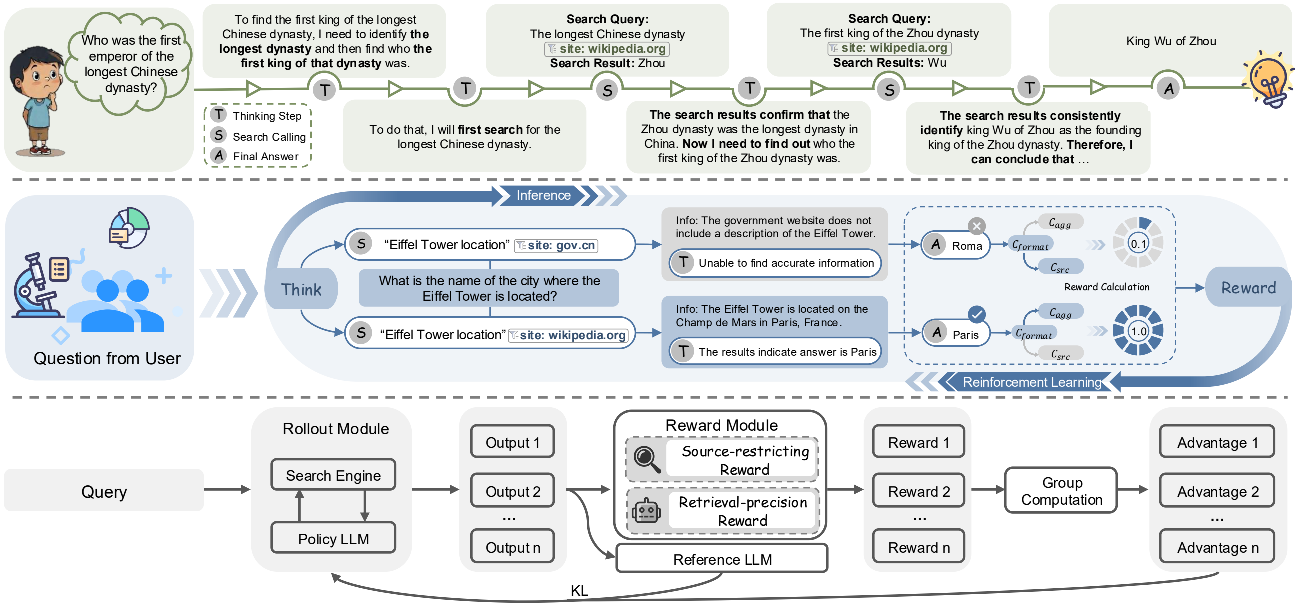


Figure 2: Overview of the WebFilter training framework. **Upper:** We formulate retrieval as a Markov Decision Process, where the model interacts with web search tools through step-by-step actions, including query generation and evidence selection. **Middle:** To improve tool usage, we provide explicit instructions and demonstrations on how to issue effective, source-aware queries. **Lower:** The policy is optimized using a behavior- and outcome-driven Information-Filtering Reward strategy, which encourages both proper tool invocation and high-quality information retrieval.

RL Training Framework

Policy Optimization In this work, we adopt the Group Relative Policy Optimization (GRPO) (Shao et al. 2024), which improves the current policy π_θ by leveraging a reference policy $\pi_{\theta_{\text{ref}}}$ and a set of rollouts generated by a old policy $\pi_{\theta_{\text{old}}}$. To support search engine calls, the GRPO objective is extended as follows:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}A_i,\right.$$
$$\left.\text{clip}\left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)},\,1-\epsilon,\,1+\epsilon\right)A_i\right)$$
$$-\,\beta\,\mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\theta_{\text{ref}}}\right)\Bigg]. \tag{9}$$

Here, x denotes an input sampled from the data distribution $\mathcal{D}$, and y_i represents a trajectory generated by $\pi_{\theta_{\text{old}}}$. $\mathbb{D}_{\text{KL}}$ is the estimated KL divergence (Shao et al. 2024), and ϵ, β are hyperparameters that control the trust region and regularization strength, respectively. The reward r_i for each y_i is computed jointly across trajectories:

$$r_1, r_2, \ldots, r_G = R(y_1, y_2, \ldots, y_G), \tag{10}$$

and the advantage A_i is normalized within the batch as:

$$A_i = \frac{r_i - \text{mean}(r_1, r_2, \ldots, r_G)}{\text{std}(r_1, r_2, \ldots, r_G)}. \tag{11}$$

This objective encourages stable policy improvement, enabling effective integration of retrieval-based reasoning into the learning process.

Experiments

Benchmarks

Our experimental setting is built on question answering datasets that assess reasoning and retrieval capabilities in diverse scenarios. For in-domain evaluation, we use the development sets of Natural Questions (NQ) (Kwiatkowski et al. 2019), TriviaQA (TQ) (Joshi et al. 2017), HotpotQA (Yang et al. 2018), and 2Wiki (Ho et al. 2020). For out-of-domain evaluation, we include the complex open-domain reasoning dataset MuSiQue (Trivedi et al. 2022) and the web-search-focused benchmark Bamboogle (Press et al. 2022), which differ in question style and information distribution. To ensure a balanced and consistent evaluation across datasets, we select a fixed number of examples from each. Specifically, 512 examples are chosen from the development sets of NQ, TQ, HotpotQA, 2Wiki, and MuSiQue, and all 125 examples are selected from the development set of Bamboogle.

Baselines

To evaluate WebFilter's effectiveness, we compare it against several baselines representing different methodologies:
Direct Reasoning: Models relying solely on internal knowledge, such as Qwen3-32B (Yang et al. 2025a), Gemini-2.0-Flash (Team et al. 2023), and GPT-4o (Hurst et al. 2024).

Table 1: Performance of different methods on in-domain datasets, evaluated with rule-based (ACC_R) and LLM-based (ACC_L) metrics. Best results are highlighted in bold.

Environment	Method	NQ		TQ		HotpotQA		2Wiki	
		ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L
Direct Reasoning	Qwen3-32B	16.5	36.3	11.6	52.9	29.9	19.1	26.5	19.9
	Gemini-2.0-Flash	20.7	47.1	16.2	68.9	42.4	29.9	32.5	23.8
	GPT-4o	23.1	53.9	17.6	73.2	50.2	37.1	38.6	30.5
Local RAG	Search-o1	34.5	57.4	52.6	61.1	31.6	40.8	28.6	32.8
	Search-r1-base	45.4	60.0	71.9	76.2	55.9	63.0	44.6	47.9
	Search-r1-instruct	33.1	49.6	44.7	49.2	45.7	52.5	43.4	48.8
Web Search	R1-Searcher	35.4	52.3	73.1	79.1	44.8	53.1	59.4	65.8
	DeepResearcher	39.6	61.9	**78.4**	85.0	52.8	64.3	59.7	66.6
	WebFilter (Ours)	**40.1**	**63.1**	77.5	**85.4**	**55.1**	**65.2**	**60.1**	**67.5**

Local RAG: Methods retrieving knowledge from offline documents. For example, Search-o1 (Li et al. 2025a) performs multi-step reasoning by generating search queries and using the retrieved snippets as context, while Search-r1-base (Jin et al. 2025) retrieves evidence from Wikipedia during both training and inference. Search-r1-instruct (Jin et al. 2025) differs by initializing the actor with an instruct-tuned language model to guide the retrieval process.

Web Search: Methods utilizing online tools. Both our approach and R1-Searcher (Song et al. 2025) rely on the Google API for web search. In addition to Google search, DeepResearcher (Zheng et al. 2025) integrates a WebBrowser tool for web navigation, which leads to increased time spent browsing and accessing websites, thereby slowing down the overall training speed. All methods, including ours, employ Qwen-2.5-7B-instruct (Yang et al. 2024) for model inference.

Metrics

We evaluate model performance using both rule-based (ACC_R) and LLM-based (ACC_L) metrics. The rule-based metric uses an F1 score to measure overlap between predictions and reference answers, reflecting factual precision. For ACC_L, we adopt the LLM-as-Judge framework (Zheng et al. 2023), where GPT-4o-mini (Hurst et al. 2024) assesses whether model answers align semantically with the references, thus capturing nuances beyond exact matching.

Implementation Details

We implement our model using the VeRL framework (Sheng et al. 2024) and adopt Qwen2.5-7B-Instruct (Yang et al. 2024) as the backbone. The hyperparameters for the aggregation function are set as $\alpha = 0.4$ and $\beta = 0.2$. The learning rate is set to 1e-5, and training proceeds with a mini-batch size of 4,096. Each iteration processes 256 samples, generating 16 rollouts per sample. Additionally, we apply a sampling temperature of 1.0 and limit the maximum retrieval count to 10. We apply loss masking to update only model-generated tokens. Our Retrieval-precision Reward uses Qwen3-30B-A3B (Yang et al. 2025a) as the judge model, which is free and can be deployed locally.

Table 2: Performance of methods on out-of-domain datasets.

Method	Musique		Bamboogle	
	ACC_R	ACC_L	ACC_R	ACC_L
Qwen3-32B	10.7	4.9	24.7	18.4
Gemini-2.0-Flash	11.4	6.1	36.5	28.0
GPT-4o	22.5	15.0	52.6	43.2
Search-o1	16.8	21.3	46.6	53.6
Search-r1-base	26.7	27.5	56.6	57.6
Search-r1-instruct	26.5	28.3	45.0	47.2
R1-Searcher	22.8	25.6	64.8	65.6
DeepResearcher	**27.1**	29.3	71.0	72.8
WebFilter (Ours)	24.5	**30.0**	**73.1**	**74.3**

Results on In-Domain Settings

WebFilter achieves state-of-the-art performance across all four in-domain datasets, demonstrating notable strengths in multi-hop reasoning tasks, as shown in Tab. 1. On HotpotQA, it outperforms DeepResearcher by 2.3% in ACC_R, despite DeepResearcher relying on a browser tool with higher latency for broader web exploration. This advantage arises from WebFilter's ability to formulate precise, source-restricted queries using advanced search operators, effectively reducing noise in retrieved documents. Compared to local RAGs, the performance gap on 2Wiki becomes more pronounced, with WebFilter achieving around a 17% higher ACC_R through selective access to trusted external sources. These improvements reflect deliberate design choices. Unlike methods limited to fixed domains, such as R1-Searcher, which restricts access to Wikipedia, or those reliant on extensive web browsing, which introduces higher latency (e.g., DeepResearcher), WebFilter focuses on generating precise, source-restricted queries for unrestricted Google API searches. This strategy strikes a balance between retrieval flexibility and high precision, minimizing irrelevant content and enhancing the quality of retrieved contexts. As indicated by gains in ACC_L, WebFilter retrieves evidence that aligns more closely with ground-truth answers, thereby supporting stronger reasoning.

Table 3: Performance of different WebFilter variants across in-domain datasets (NQ, TQ, HotpotQA, 2Wiki) and out-of-domain datasets (Musique, Bamboogle). “SR” denotes the Source-restricting Reward, and “RR” denotes the Retrieval-precision Reward.

Methods	**NQ**		**TQ**		**HotpotQA**		**2Wiki**		**Musique**		**Bamboogle**	
	ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L	ACC_R	ACC_L
Base	41.2	63.6	78.5	82.6	49.4	59.9	55.2	58.2	22.7	26.2	64.9	65.8
Base+SR	**41.4**	**64.3**	**79.0**	**86.0**	50.6	60.1	59.7	65.3	24.5	27.6	64.6	65.2
Base+SR+RR(Ours)	40.1	63.1	77.5	85.4	**55.1**	**65.2**	**60.1**	**67.5**	**24.5**	**29.0**	**73.1**	**74.3**

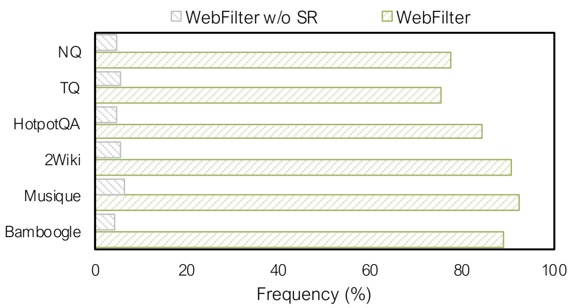
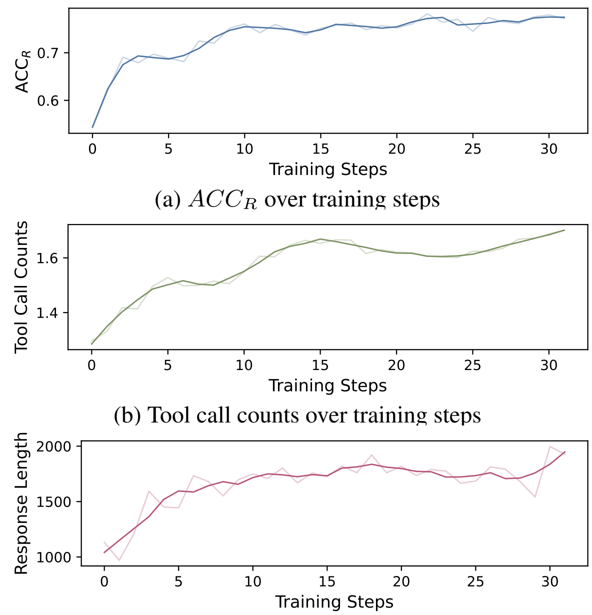


Figure 3: Frequency of advanced operators across variants.



(a) ACC_R over training steps

(b) Tool call counts over training steps

(c) Response length over training steps

Figure 4: Training dynamics showing (a) QA accuracy (ACC_R), (b) tool call behavior, and (c) response length evolution across training steps.

Results on Out-of-Domain Settings

WebFilter consistently shows strong generalization on out-of-domain datasets. As shown in Tab. 2, WebFilter achieves the highest ACC_L on the challenging open-domain Musique dataset (30.0%) and the web-search-heavy Bamboogle dataset (74.3%). These results suggest that WebFilter retrieves evidence more semantically aligned with ground-truth answers, which is critical for reasoning beyond exact matching. WebFilter also maintains competitive ACC_R scores, particularly on Bamboogle (73.1%), indicating its ability to preserve factual consistency in new domains. Moreover, it outperforms DeepResearcher on ACC_L for Musique (30.0% vs. 29.3%), demonstrating its strength in handling difficult open-domain questions. WebFilter's ability to retrieve semantically relevant and factually consistent information across diverse topics underscores its practical value for real-world applications involving domain shifts and open-ended queries.

Ablation Study

Tab. 3 shows the performance of WebFilter variants on both in-domain datasets (NQ, TQ, HotpotQA, 2Wiki) and out-of-domain datasets (Musique, Bamboogle). From the results, we observe that incorporating the Source-restricting Reward (SR) leads to noticeable gains on in-domain datasets, improving the model's ability to retrieve information from reliable sources. SR also encourages the use of advanced operators. As shown in Fig. 3, SR-guided queries include advanced operators more than 75% of the time, compared to less than 10% without SR. Adding the Retrieval-precision Reward (RR) further boosts performance, especially on out-of-domain datasets. RR refines the retrieval process, aligning the generated evidence more closely with ground-truth answers. For instance, on Bamboogle, ACC_R improves by 8.5%, from 64.6% to 73.1%, and ACC_L rises from 65.2% to 74.3%. This improvement is mainly due to Bamboogle's focus on web-search-based reasoning, where WebFilter's ability to leverage advanced search operators and filter irrelevant content enhances retrieval of high-quality, relevant evidence. The combination of SR and RR yields the best results, with SR boosting retrieval precision and RR enhancing generalization. These findings demonstrate the effectiveness of our reward framework in improving both retrieval quality and reasoning performance.

Training Dynamics Analysis

Fig. 4 illustrates the training dynamics of our model, focusing on both QA performance and behavioral changes. As shown in Fig. 4a, ACC_R steadily increases throughout training, rising from approximately 0.52 to around 0.77 on the TQ dataset. This indicates continuous improvements in retrieval accuracy as the model progresses.

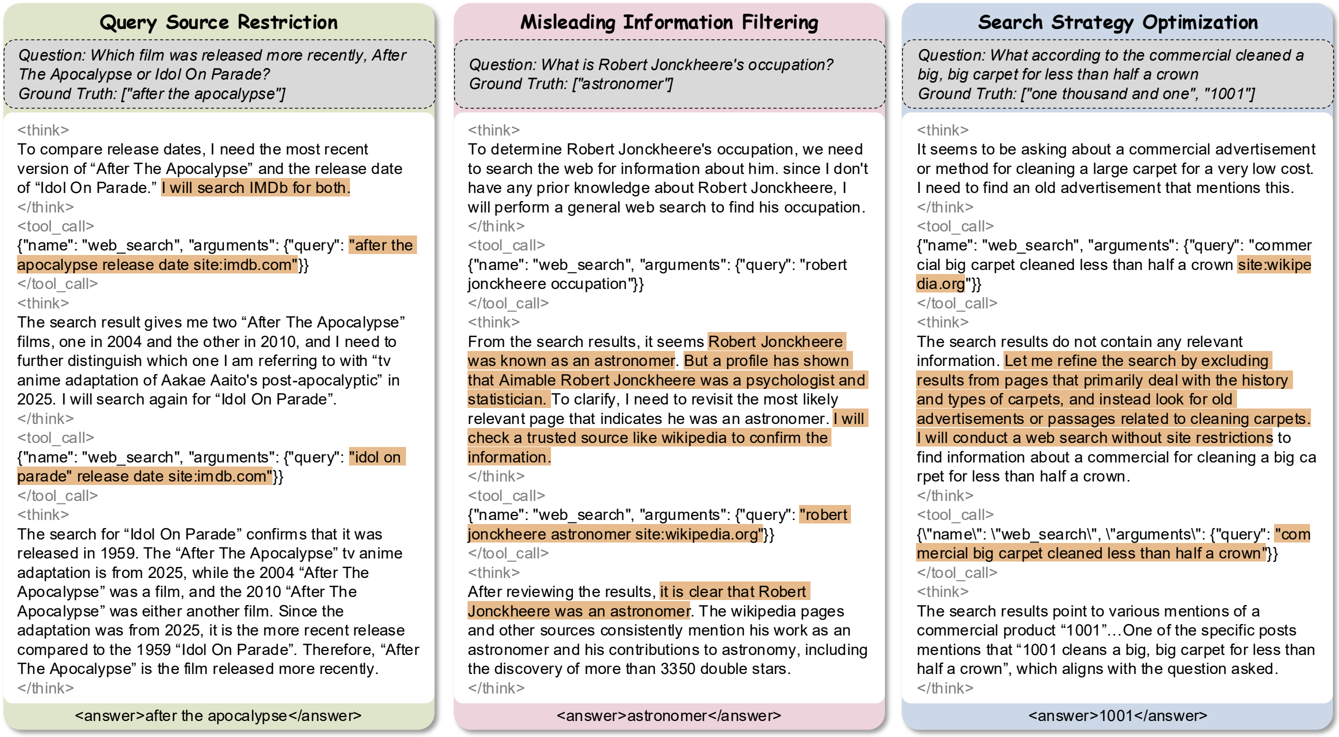


Figure 5: Case studies showing how WebFilter improves QA by (a) narrowing searches to authoritative sources for precise results, (b) verifying ambiguous or conflicting information via trusted sites, and (c) adaptively refining search queries when initial attempts are insufficient.

To capture broader trends, Fig. 4b and Fig. 4c show the average tool call counts and response lengths across all evaluated datasets. In Fig. 4b, tool call counts increase during early training, plateau around step 20, and then rise again. This pattern suggests that the model gradually incorporates more frequent retrieval as training progresses. Meanwhile, Fig. 4c reveals that the average response length grows from about 1,000 tokens to nearly 2,000 tokens, indicating that the model generates more detailed and comprehensive responses over time. Overall, these results demonstrate that our model not only improves retrieval accuracy but also adapts its tool usage and response strategies, leading to more effective and informative outputs.

Case Study

We present three representative cases in Fig. 5 to illustrate WebFilter's intelligent retrieval behaviors. Through systematic analysis, we identify three key behavioral patterns that define the model's advanced search capabilities:

Query Source Restriction. For domain-specific queries, WebFilter proactively limits search scopes to authoritative sources. In Case 1, when asked about film release dates, it automatically appends "site:imdb.com" to queries. This targeted approach ensures precise, trustworthy results while reducing latency by filtering out irrelevant information.

Misleading Information Filtering. WebFilter demonstrates strong disambiguation skills for resolving conflicting information. In Case 2, when the initial search reveals both an astronomer and a similarly named psychologist, the model detects the ambiguity and performs a refined follow-up search limited to Wikipedia, successfully verifying the correct occupation.

Search Strategy Optimization. WebFilter dynamically adjusts its retrieval strategy when initial searches are insufficient. In Case 3, failing to find relevant information via a Wikipedia-restricted query, the model expands its search without site constraints, ultimately locating references to the product "1001," which matches the question context.

These cases highlight WebFilter's ability to reason about search scope, verify information across sources, and adapt strategies to improve retrieval effectiveness.

Limitations

While our approach shows progress, it has some limitations. To enhance RAG capabilities, improving search quality is crucial, but not sufficient on its own. In many cases, errors occur not because the model fails to retrieve relevant information, but because it struggles to correctly interpret and reason with the retrieved data. Our model, when combined with improvements in reasoning abilities, can deliver even greater value in future work.

Conclusion

We present WebFilter, a framework that improves Retrieval-Augmented Generation (RAG) by leveraging advanced search operators for precise, source-aware queries and misinformation filtering. By modeling retrieval as a Markov Decision Process, WebFilter learns to effectively use web search tools. Experiments demonstrate strong gains on both in-domain and out-of-domain QA. Future work will explore broader web interactions to further enhance real-world RAG performance.

Acknowledgements

This work was supported by the Ant Group Research Intern Program.

References

Casper, S.; Davies, X.; Shi, C.; Gilbert, T. K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Chen, M.; Li, T.; Sun, H.; Zhou, Y.; Zhu, C.; Wang, H.; Pan, J. Z.; Zhang, W.; Chen, H.; Yang, F.; et al. 2025. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Hao, S.; Gu, Y.; Ma, H.; Hong, J. J.; Wang, Z.; Wang, D. Z.; and Hu, Z. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Ho, X.; Nguyen, A.-K. D.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12): 1–38.

Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arik, S.; Wang, D.; Zamani, H.; and Han, J. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4: 237–285.

Kaufmann, T.; Weng, P.; Bengs, V.; and Hüllermeier, E. 2023. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 10.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Li, X.; Dong, G.; Jin, J.; Zhang, Y.; Zhou, Y.; Zhu, Y.; Zhang, P.; and Dou, Z. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

Li, X.; Jin, J.; Dong, G.; Qian, H.; Zhu, Y.; Wu, Y.; Wen, J.-R.; and Dou, Z. 2025b. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. *arXiv preprint arXiv:2504.21776*.

Li, X.; Zou, H.; and Liu, P. 2025. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Pang, R. Y.; Yuan, W.; He, H.; Cho, K.; Sukhbaatar, S.; and Weston, J. 2024. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37: 116617–116637.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Qi, Z.; Liu, X.; Iong, I. L.; Lai, H.; Sun, X.; Zhao, W.; Yang, Y.; Yang, X.; Sun, J.; Yao, S.; et al. 2024. WebRL: Training LLM Web Agents via Self-Evolving Online Curriculum Reinforcement Learning. *arXiv preprint arXiv:2411.02337*.

Qiu, J.; Lam, K.; Li, G.; Acharya, A.; Wong, T. Y.; Darzi, A.; Yuan, W.; and Topol, E. J. 2024. LLM-based agentic systems in medicine and healthcare. *Nature Machine Intelligence*, 6(12): 1418–1420.

Sahoo, S. S.; Plasek, J. M.; Xu, H.; Uzuner, Ö.; Cohen, T.; Yetisgen, M.; Liu, H.; Meystre, S.; and Wang, Y. 2024. Large language models for biomedicine: foundations, opportunities, challenges, and best practices. *Journal of the American Medical Informatics Association*, 31(9): 2114–2124.

Sha, Z.; Cui, S.; and Wang, W. 2025. SEM: Reinforcement Learning for Search-Efficient Large Language Models. *arXiv preprint arXiv:2505.07903*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sheng, G.; Zhang, C.; Ye, Z.; and ... 2024. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv:2409.19256*.

Singh, J.; Magazine, R.; Pandya, Y.; and Nambi, A. 2025. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*.

Song, H.; Jiang, J.; Min, Y.; Chen, J.; Chen, Z.; Zhao, W. X.; Fang, L.; and Wen, J.-R. 2025. R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2503.05592*.

Tang, F.; Gu, Z.; Lu, Z.; Liu, X.; Shen, S.; Meng, C.; Wang, W.; Zhang, W.; Shen, Y.; Lu, W.; et al. 2025. GUI-G: Gaussian Reward Modeling for GUI Grounding. *arXiv preprint arXiv:2507.15846*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Wang, H.; Qian, C.; Zhong, W.; Chen, X.; Qiu, J.; Huang, S.; Jin, B.; Wang, M.; Wong, K.-F.; and Ji, H. 2025. Otc: Optimal tool calls via reinforcement learning. *arXiv e-prints*, arXiv–2504.

Wei, Z.; Yao, W.; Liu, Y.; Zhang, W.; Lu, Q.; Qiu, L.; Yu, C.; Xu, P.; Zhang, C.; Yin, B.; Yun, H.; and Li, L. 2025. WebAgent-R1: Training Web Agents via End-to-End Multi-Turn Reinforcement Learning. arXiv:2505.16421.

Xie, Z.; Cao, J.; Zhang, Y.; Zhang, Q.; and Xu, R. 2025. A Dual-Agent Adversarial Framework for Robust Generalization in Deep Reinforcement Learning. arXiv:2501.17384.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115, 2024*.

Yang, S.; Dai, Y.; Wang, G.; Zheng, X.; Xu, J.; Li, J.; Ying, Z.; Wang, W.; and Ngai, E. C. H. 2025b. RealFactBench: A Benchmark for Evaluating Large Language Models in Real-World Fact-Checking. arXiv:2506.12538.

Yang, S.; Yu, Z.; Ying, Z.; Dai, Y.; Wang, G.; Lan, J.; Xu, J.; Li, J.; and Ngai, E. C. H. 2025c. RAMA: Retrieval-Augmented Multi-Agent Framework for Misinformation Detection in Multimodal Fact-Checking. arXiv:2507.09174.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yu, Y.-C.; Chiang, T.-H.; Tsai, C.-W.; Huang, C.-M.; and Tsao, W.-K. 2025. Primus: A pioneering collection of open-source datasets for cybersecurity LLM training. *arXiv preprint arXiv:2502.11191*.

Zhang, S.; Dong, Y.; Zhang, J.; Kautz, J.; Catanzaro, B.; Tao, A.; Wu, Q.; Yu, Z.; and Liu, G. 2025. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. 2023. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhang, Y.; Sharma, K.; Du, L.; and Liu, Y. 2024. Toward mitigating misinformation and social media manipulation in llm era. In *Companion Proceedings of the ACM Web Conference 2024*, 1302–1305.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zheng, Y.; Fu, D.; Hu, X.; Cai, X.; Ye, L.; Lu, P.; and Liu, P. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.