# SmartSearch: Process Reward-Guided Query Refinement for Search Agents

Tongyu Wen
Renmin University of China
Beijing, China
wentongyu@ruc.edu.cn

Guanting Dong
Renmin University of China
Beijing, China
dongguanting@ruc.edu.cn

Zhicheng Dou*
Renmin University of China
Beijing, China
dou@ruc.edu.cn

## Abstract

Large Language Model (LLM)-based search agents have proven promising for addressing knowledge-intensive problems by incorporating information retrieval capabilities. Existing works largely focus on optimizing the reasoning paradigms of search agents, yet the quality of intermediate search queries during reasoning remains overlooked. As a result, the generated queries often remain inaccurate, leading to unexpected retrieval results and ultimately limiting search agents' overall effectiveness. To mitigate this issue, we introduce **SmartSearch**, a framework built upon two key mechanisms: (1) **Process Rewards**, which provide fine-grained supervision for the quality of each intermediate search query through Dual-Level Credit Assessment. (2) **Query Refinement**, which promotes the optimization of query generation by selectively refining low-quality search queries and regenerating subsequent search rounds based on these refinements. To enable the search agent to progressively internalize the ability to improve query quality under the guidance of process rewards, we design a three-stage curriculum learning framework. This framework guides the agent through a progression from imitation, to alignment, and ultimately to generalization. Experimental results show that SmartSearch consistently surpasses existing baselines, and additional quantitative analyses further confirm its significant gains in both search efficiency and query quality. The code is available at https://github.com/RUC-NLPIR/SmartSearch.

## CCS Concepts

• **Information systems** → **Information retrieval**; **Language models**; **Question answering**.

## Keywords

Search Agent, Information Retrieval, Large Language Models

**ACM Reference Format:**
Tongyu Wen, Guanting Dong, and Zhicheng Dou. 2026. SmartSearch: Process Reward-Guided Query Refinement for Search Agents. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3805712.3809700

*Corresponding author


*SIGIR '26, Melbourne, VIC, Australia*

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809700

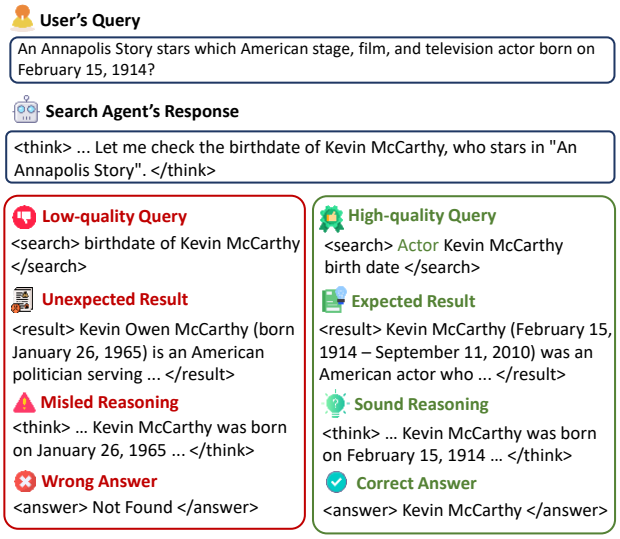


**Figure 1: An example from ASearcher [14] dataset demonstrating how low-quality intermediate search queries lead to unexpected retrieval results and derail the entire trajectory.**

## 1 Introduction

Large Language Models (LLMs) have shown strong performance across a variety of tasks [1, 2, 6, 44, 45], including translation [56, 62], summarization [40, 64], and question answering [23, 37]. However, challenges remain, particularly with issues like hallucinations [17, 34] and the absence of recent or field-specific knowledge, which may result in inaccurate or outdated answers. Retrieval-augmented generation (RAG) [3, 15, 19] has been introduced to address these challenges by incorporating external knowledge to complement the model's internal knowledge [35, 52]. However, static RAG often faces limitations in its ability to handle more complex, dynamic, and deep exploration tasks.

Recently, LLM-based search agents have proven to be a promising method [4, 21, 27, 28, 38, 39]. These agents can autonomously and iteratively invoke external search tools, thereby addressing more challenging knowledge-intensive problems that demand adaptive retrieval and in-depth reasoning. Current research on search agents has made considerable progress in optimizing the reasoning paradigms of search agents through methods like prompt engineering [27] and fine-tuning [4, 21, 38, 39]. However, they often overlook the quality of intermediate search queries during reasoning, yet low-quality queries can lead to unexpected retrieval results or even

derail the entire trajectory. Figure 1 illustrates how minor inaccuracies in an intermediate search query (e.g., omitting 'actor') can lead a search agent to retrieve and accept unexpected information, ultimately resulting in an incorrect answer. This highlights the critical role that search query quality plays in the deep information-seeking process. Some studies [9, 57, 65, 67] have attempted to incorporate process rewards into search agent training. However, they tend to focus more on shaping better reasoning behavior rather than improving the quality of intermediate search queries, and existing efforts [67] on intermediate search queries remain preliminary and ineffective. Furthermore, research [20, 41] has shown that existing training paradigms often prioritize information utilization, persistently neglecting the optimization of retrieval patterns. This undoubtedly impedes the search agent's ability to achieve deep and reliable information retrieval, thereby compromising its overall effectiveness. Such issues highlight the need for methods that specifically focus on optimizing query quality during training.

In this work, we present **SmartSearch**, a framework that optimizes search query quality through the guidance of process rewards, thereby enhancing the deep information-seeking capabilities of search agents. Specifically, SmartSearch incorporates two key mechanisms to provide step-level feedback on intermediate search queries and promote the optimization of query generation. **(1) Process Rewards**: To provide fine-grained supervision for the quality of each intermediate search query, we introduce Dual-Level Credit Assessment, which comprises two complementary components. The first one is a rule-based assessment for query novelty, which detects redundancy by checking whether the retrieved documents contain excessive overlap with previous rounds. The second one is a model-based evaluation for query usefulness, which judges whether the query intent is necessary and whether the retrieved results provide the expected answer. This mechanism outputs both numerical scores and textual feedback, which serve as guidance for subsequent query refinement. **(2) Query Refinement**: To further promote the optimization of query generation during training, the agent first generates a complete search trajectory, then identifies low-quality search rounds according to the numerical scores from the process rewards. Subsequently, we employ a model to refine those queries under the textual guidance provided by the process rewards, after which the search agent continues generating from the refined queries. To improve the efficiency of query assessment and refinement, a smaller model is trained for scoring and refinement, reducing computational cost while maintaining effectiveness.

Building on the foundation of the two mechanisms, we introduce a three-stage curriculum learning framework. The framework guides the search agent through a progression from imitation and alignment to generalization, enabling it to progressively internalize the ability to enhance query quality under the guidance of process rewards. **(1) Query Quality Screened Imitation Learning**: The initial stage leverages Supervised Fine-Tuning (SFT) to guide the search agent during its early learning of information retrieval and utilization. The training data is filtered based on both final answer correctness and query quality measured by the process rewards. It ensures the model to learn from trajectories that not only lead to correct answers but also maintain high-quality search processes. **(2) Query Generation Alignment**: In this stage, the search agent cultivates advanced query generation capabilities through Direct Preference Optimization (DPO). We employ the query refinement mechanism to generate comparative data, with process rewards and outcome rewards jointly defining which trajectories are of higher quality. **(3) Query-Aware Policy Optimization**: The final stage utilizes Reinforcement Learning (RL) to further strengthen its integrated capabilities of information retrieval and utilization. During the rollout phase, the query refinement mechanism is employed, with the process rewards incorporated into the reward function.

To thoroughly assess the capabilities of SmartSearch, we perform experiments on four challenging knowledge-intensive tasks and two web exploration tasks. Experimental results indicate that SmartSearch consistently surpasses all baselines in overall performance and exhibits strong generalization to open-web settings. Additionally, we perform a range of ablation studies and quantitative analyses to comprehensively validate SmartSearch's effectiveness. Our findings highlight the critical contribution of our two key mechanisms and three curriculum learning stages, as well as their superiority in terms of search efficiency, search query quality, and other dimensions.

To summarize, the primary contributions of this study include:

(1) We present a pioneering focus that optimizes the quality of intermediate search queries through process reward guidance, thereby improving the information-seeking ability of search agents.

(2) We propose SmartSearch, a framework that incorporates two key mechanisms: process rewards and query refinement, to enable process reward-guided search refinement.

(3) We design a three-stage, query-oriented curriculum learning framework that guides the agent through a progression from imitation and alignment to generalization, progressively internalizing the ability to improve query quality.

(4) Experiments across six challenging benchmarks demonstrate that SmartSearch consistently surpasses existing baselines, and further quantitative analyses confirm significant improvements in both search efficiency and query quality.

## 2 Related Works

### 2.1 LLM-based Search Agents

LLMs have demonstrated strong performance across various tasks [1, 2, 6, 44, 45], yet challenges like hallucinations [17, 34] and static parametric knowledge remain. While the RAG [3, 15, 19] paradigm addresses these issues by incorporating external knowledge, it still struggles with complex, dynamic, and deep exploration tasks. Recently, LLM-based search agents have emerged as a promising solution [4, 21, 27, 28, 38, 39]. This advanced paradigm enables models to autonomously and iteratively invoke external tools, effectively tackling challenging knowledge-intensive problems.

Research on search agents has progressed through methods including prompt engineering and fine-tuning. Early prompt-based methods [27, 30, 47] focused on carefully designed prompts and structured workflows to steer the agent's behavior. However, these methods don't fundamentally enhance the model's underlying capabilities, leading many studies to shift towards fine-tuning-based approaches. A prominent line of work has demonstrated that SFT [12, 13, 18, 29] on expert trajectories enables agents to learn through imitation and yields promising performance. Building upon this foundation, recent studies [4, 10, 21, 38, 39] have

employed RL to further advance search agent capabilities. However, existing methods tend to overlook intermediate search query quality, which can lead to unexpected retrieval results or even derail the entire trajectory. Moreover, research [20] indicates that current training paradigms tend to prioritize information utilization, which can lead to stagnation in information retrieval abilities. Thus, we present a framework designed to optimize the quality of intermediate search queries under the guidance of process rewards, thereby enhancing the overall performance of search agents.

### 2.2 Process Rewards in RL

Recent advancements in RL have achieved significant success in large reasoning models [7, 42, 43], and have also demonstrated effectiveness in enhancing the performance of LLM-based search agents [4, 10, 21, 27, 28, 38, 39]. However, reward signals based solely on final outcomes often result in sparse feedback in multi-round search tasks, providing insufficient guidance for intermediate steps and leading to unstable and inefficient policy optimization [63]. To overcome this limitation, recent studies [9, 57, 65, 67] have explored the use of process-based rewards. Some approaches employ Monte Carlo Tree Search to estimate intermediate actions' value [25, 65], while others rely on a corpus of annotated golden steps or intermediate information to compute rewards based on alignment with this reference [48, 50, 61, 66, 67]. Still others leverage external reward models to provide fine-grained evaluation for each step [9, 54, 55, 57, 58].

These approaches have proven effective in enhancing the effectiveness and stability of RL training. Yet, most of these approaches tend to focus primarily on the quality of the reasoning process rather than the quality of intermediate search queries [9, 58], with existing efforts on intermediate search queries remaining preliminary and ineffective [67]. In this context, our process rewards mechanism provides fine-grained supervision for query quality through Dual-Level Credit Assessment, playing a central role in the query-oriented training framework.

## 3 Preliminaries

### 3.1 Task Formulation

We adopt ReAct [60] as the framework for the search agent. Given a user query $q$, the search agent, guided by an LLM policy $\pi_\theta$, interacts with an external search tool through several iterations of Thought-Action-Observation to gather information and ultimately generate an answer. Specifically, during each iteration, the search agent starts by engaging in thinking to generate a "Thought" according to the existing context. It then produces the next "Action", which involves querying the search tool. The agent subsequently waits for the environment to return the "Observation", consisting of the Top-K retrieved document fragments for the search query. The iteration concludes when the search agent has gathered sufficient information required to address the user's question and selects the "final answer" as the action. A complete trajectory over $T$ iterations is denoted as:

$$H_T = (q, \tau_0, a_0, o_0, \ldots, \tau_i, a_i, o_i, \ldots, \tau_T, a_T). \tag{1}$$

Here, $\tau_i$, $a_i$, and $o_i$ correspond to the Thought, Action, and Observation of the $i$-th iteration. In iteration $t$, the LLM policy $\pi_\theta(a, t|H_{t-1})$ produces the thought $\tau_t$ and action $a_t$, which is conditioned on the entire history of prior context $H_{t-1}$.

Following prior work [10], we use the prompt as shown below for SmartSearch.

**Prompt for SmartSearch.**

You are a helpful assistant that can solve the given question step by step with the help of the wikipedia search tool. Given a question, you need to first think about the reasoning process in the mind and then provide the answer. During thinking, you can invoke the wikipedia search tool to search for fact information about specific topics if needed. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags respectively, and the search query and result are enclosed within <search> </search> and <result> </result> tags respectively. For example, <think> This is the reasoning process. </think> <search> search query here </search> <result> search result here </result> <think> This is the reasoning process. </think> <answer> The final answer is

answer here

</answer>. In the last part of the answer, the final exact answer is enclosed within \boxed{ } with latex format.

### 3.2 Agentic Reinforcement Learning

***Policy Optimization.*** In the context of Agentic RL [63], Group Relative Policy Optimization (GRPO) [36] is typically employed for policy optimization. In our approach, we also employ GRPO during the Query Aware Policy Optimization stage, with a specific focus on the augmentation of the rollout and reward modules to optimize the quality of intermediate search queries. Specifically, GRPO optimizes the policy model through maximization of the objective function below:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_t(\theta)\hat{A}_i,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big) - \beta\, D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right]. \tag{2}$$

In this formulation, for each input pair $(q, a)$ drawn from the dataset $\mathcal{D}$, $G$ trajectories $\{o_i\}_{i=1}^{G}$ are generated from the old policy $\pi_{\theta_{\text{old}}}(\cdot \mid q)$. Here, $\epsilon$ is a small hyperparameter that defines the clipping range $[1-\epsilon, 1+\epsilon]$ for the probability ratio $r_t(\theta)$, preventing excessively large policy updates and improving training stability. Additionally, $\pi_{\text{ref}}$ is a fixed reference policy typically initialized as the policy before GRPO updates, and the term $-\beta\, D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})$ acts as a regularizer that penalizes the new policy $\pi_\theta$ from deviating too far from $\pi_{\text{ref}}$, thereby mitigating catastrophic forgetting and ensuring stable learning. The importance weight $r_t(\theta)$ is defined as the following:

$$r_t(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{3}$$

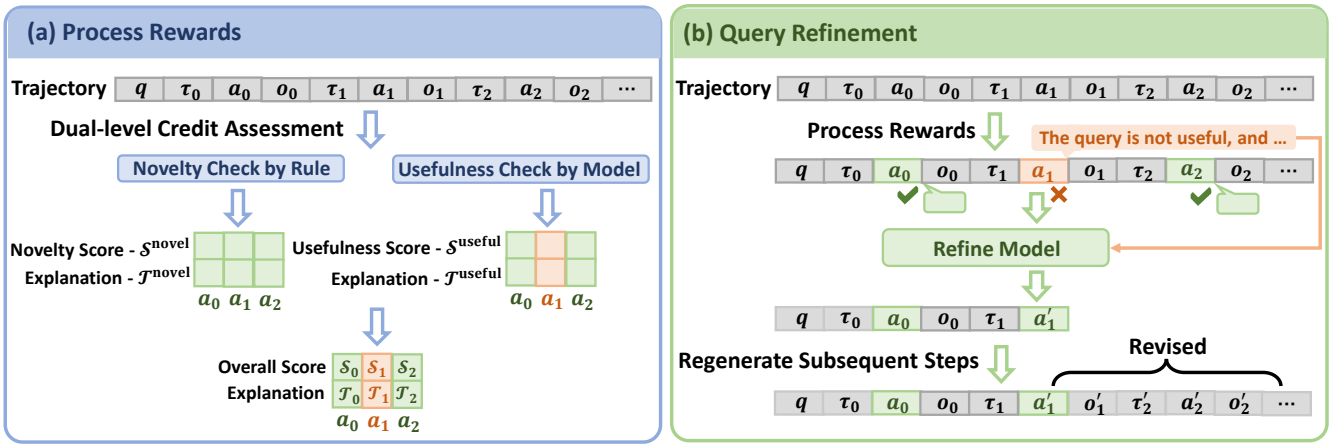


**Figure 2: An overview of the two key mechanisms in SmartSearch: the process rewards (a) and the query refinement (b).**

The normalized advantage score $\hat{A}_i$ is denoted as:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^{G})}{\text{std}(\{r_j\}_{j=1}^{G})}. \tag{4}$$

Here, $r_i$ denotes the scalar reward for the $i$-th rollout. Furthermore, agentic RL typically masks observations originating from the external environment during loss computation, thereby effectively preventing unstable training.

***Reward Design.*** As discussed earlier, in agentic RL, each rollout corresponds to a scalar reward $r$. Prior research [4, 21, 38, 39] predominantly relies on combining two key types of rewards: the outcome reward $r_{\text{outcome}}$, reflecting the trajectory's answer correctness, and the format reward $r_{\text{format}}$, assessing the trajectory's structural correctness. These rewards are typically weighted and combined using a simple hyperparameter $\lambda$ as follows:

$$r = r_{\text{outcome}} + \lambda \cdot r_{\text{format}}. \tag{5}$$

In some recent works [9, 57, 58], process rewards have been incorporated into the reward function to provide fine-grained feedback on intermediate steps. The reward function is then extended to include the process rewards, with a composite reward incorporating both the outcome reward and the process rewards, while the format reward is weighted by a hyperparameter:

$$r = r_{\text{composite}} + \lambda \cdot r_{\text{format}}. \tag{6}$$

Here, $r_{\text{composite}}$ is computed as the aggregation of multiple step-wise process rewards and the final outcome reward $r_{\text{outcome}}$:

$$r_{\text{composite}} = f(r_1^{\text{process}}, r_2^{\text{process}}, \ldots, r_n^{\text{process}}, r_{\text{outcome}}), \tag{7}$$

where $n$ represents the total steps in the trajectory, and $r_i^{\text{process}}$ denotes the process reward for the $i$-th step. The aggregation function $f$ combines these individual rewards, and its specific form may vary across different works.

# 4 Our Method

## 4.1 Overview

We propose SmartSearch, a framework that enhances search agent performance by optimizing the quality of intermediate search queries through process reward guidance. As illustrated in Figure 2, SmartSearch incorporates two key mechanisms: (1) **Process Rewards** (§4.2), which provide fine-grained supervision for the quality of each intermediate search query through Dual-Level Credit Assessment. (2) **Query Refinement** (§4.3), which promotes the optimization of query generation by selectively refining low-quality queries and regenerating subsequent search rounds based on these refinements. To further internalize the ability to improve query quality, we propose a three-stage curriculum learning framework (§4.4) built upon these mechanisms. As shown in Figure 3, it comprises **Query Quality Screened Imitation Learning**, **Query Generation Alignment**, and **Query Aware Policy Optimization**. Below, we will first introduce the two key mechanisms, followed by a detailed description of the three-stage curriculum learning framework.

## 4.2 Process Reward for Assessing Query Quality

In this section, we introduce the process rewards mechanism to assess the quality of each intermediate search query, providing both numerical scores and textual feedback. These outputs guide the subsequent query refinement, and play a key role within the three-stage curriculum learning framework by selecting trajectories with high-quality search processes and providing finer-grained supervision signals, which will be introduced later.

***Design Principles.*** Our assessment of search query quality is guided by a comprehensive set of three fundamental principles:

- **Query Novelty**: The query should avoid redundancy with previous queries and introduce novel information.
- **Intent Necessity**: The query's search intent must be necessary for progressing *toward the final answer.*
- **Retrieval Relevance**: The retrieved documents should align with the search intent, effectively containing the expected information or answer.

These principles are well-motivated and collectively capture the essential aspects of a high-quality query, while also being readily applicable via either rule-based checks or simple model judgments.

***Dual-Level Credit Assessment.*** We operationalize these principles through Dual-Level Credit Assessment, which consists of two complementary components.

**(1) Rule-based Evaluation:** The first is a rule-based evaluation for query novelty, which identifies redundant queries by measuring the document overlap between the current and previous search rounds. Formally, for the $t$-th step, the novelty score $\mathcal{S}_t^{\text{novel}}$ and its corresponding textual explanation $\mathcal{T}_t^{\text{novel}}$ are defined as:

$$\left(\mathcal{S}_t^{\text{novel}}, \mathcal{T}_t^{\text{novel}}\right) = \begin{cases} (0, \text{the query is redundant}), & \text{if } O_t > K, \\ (1, \text{the query is novel}), & \text{if } O_t \le K. \end{cases} \tag{8}$$

Here, $K$ is a threshold hyperparameter, and $O^t$ represents the number of documents retrieved at step $t$ that share the same content with those retrieved in any previous step, defined as:

$$O^t = \sum_{i=1}^{n} \mathbb{I}(D_i^t \in \bigcup_{s=0}^{t-1} \bigcup_{j=1}^{n} D_j^s), \tag{9}$$

where $D_i^t$ refers to the $i$-th document retrieved at step $t$, and $\mathbb{I}(\cdot)$ is the indicator function.

**(2) Model-based Evaluation:** The second component is model-based evaluation for query usefulness, which assesses the necessity of the query intent and checks whether the retrieved results provide the expected answer. For the $t$-th step, the evaluation score $\mathcal{S}_t^{\text{useful}}$ and its corresponding textual explanation $\mathcal{T}_t^{\text{useful}}$ are defined as:

$$\mathcal{S}_t^{\text{useful}}, \mathcal{T}_t^{\text{useful}} = \text{LLM}_{\text{eval}}(q, a, H_t), \tag{10}$$

where $\text{LLM}_{\text{eval}}$ is the model used for evaluation, $q$ is the user's query, $a$ denotes the golden answer, and $H_t$ indicates the trajectory up to step $t$. The score $\mathcal{S}_t^{\text{useful}}$ is set to 1 if the query meets the criteria, and 0 otherwise, while the explanation $\mathcal{T}_t^{\text{useful}}$ is directly parsed from the model's output. To enhance efficiency, we employ a smaller model fine-tuned via SFT for both scoring and the subsequent query refinement task. Specifically, we input task-specific prompts into a more powerful teacher model and use its outputs as annotation labels. The smaller model is then trained on these prompt-output pairs, enabling it to achieve effective performance at a reduced computational cost. More details about the model will be introduced in Section 5.1.

Finally, the overall assessment score $\mathcal{S}_t$ and its corresponding textual explanation $\mathcal{T}_t$ are derived by aggregating the evaluations for query novelty and usefulness. The overall score is determined by a logical conjunction of the component scores:

$$\mathcal{S}_t = \begin{cases} 1, & \text{if } \mathcal{S}_t^{\text{novel}} = 1 \wedge \mathcal{S}_t^{\text{useful}} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The final explanation is synthesized by concatenating the textual feedback from both components:

$$\mathcal{T}_t = \mathcal{T}_t^{\text{novel}} \,\|\, \mathcal{T}_t^{\text{useful}}, \tag{12}$$

where $\|$ denotes the concatenation operator.

## 4.3 Process Reward-Guided Query Refinement

This section introduces the query refinement mechanism, which is designed to promote the optimization of query generation. It is achieved by systematically identifying and refining low-quality queries, then regenerating subsequent search steps from these refined points. This mechanism serves a pivotal function within the three-stage curriculum learning framework by generating comparative data for training and acting as a rollout strategy.

Formally, this process can be represented as follows. The search agent starts by generating a complete trajectory $H_T$, represented as $(q, \tau_0, a_0, o_0, \ldots, \tau_i, a_i, o_i, \ldots, \tau_T, a_T)$. Each search query in this trajectory is then evaluated by the process rewards mechanism, yielding a sequence of scores $(\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{T-1})$ and corresponding textual explanations $(\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_{T-1})$ with Equation (11) and (12). For each low-quality query $a_i$ where the score $\mathcal{S}_i = 0$, a refinement step is triggered. The refined query $a_i'$ is generated by a language model as follows:

$$a_i' = \text{LLM}_{\text{refine}}(q, H_i, \mathcal{T}_i). \tag{13}$$

Here, $\text{LLM}_{\text{refine}}$ is the same lightweight SFT-tuned model introduced earlier, $q$ is the user's original query, $H_i$ is the trajectory history up to step $i$, and $\mathcal{T}_i$ is the textual feedback diagnosing the quality issue for the low-quality query $a_i$. The search agent subsequently regenerates the search process from this refined query $a_i'$, yielding a new trajectory $H_T'$, represented as $(q, \tau_0, a_0, o_0, \ldots, \tau_i, a_i', o_i', \ldots, \tau_T', a_T')$. The primary distinction between the initial and revised trajectories originates from the refined query $a_i'$, resulting in a different reward for $a_i$ and $a_i'$, thereby promoting the optimization of query generation within the curriculum learning framework.

Specifically, to enable the model to effectively refine the low-quality search query based on the textual feedback provided by the process rewards mechanism, we distill key empirical insights into a set of practical actionable guidelines based on a thorough analysis of representative cases:

- If the textual feedback indicates that the query is redundant or unnecessary, the refined query should serve for a more necessary intent and eliminate redundancy.
- If the textual feedback indicates that the retrieved results do not contain the expected information, the model should strategically reformulate the query to better capture the target content. This reformulation may involve switching between a complete semantic question and a keyphrase-based query, or adaptively adding or removing information from the original query.

## 4.4 Query-Oriented Training Framework

This section presents a three-stage curriculum learning framework that integrates the two preceding mechanisms, enabling the agent to progressively internalize the ability to improve query quality through a progression from imitation, to alignment, and ultimately to generalization. The following paragraphs detail the three progressive stages: Query Quality Screened Imitation Learning, Query Generation Alignment, and Query Aware Policy Optimization.

***Stage-1: Query Quality Screened Imitation Learning.*** In this stage, we employ SFT to guide the model in its initial learning of information retrieval and utilization. A critical step in SFT is the selection of high-quality trajectories for training. Following common

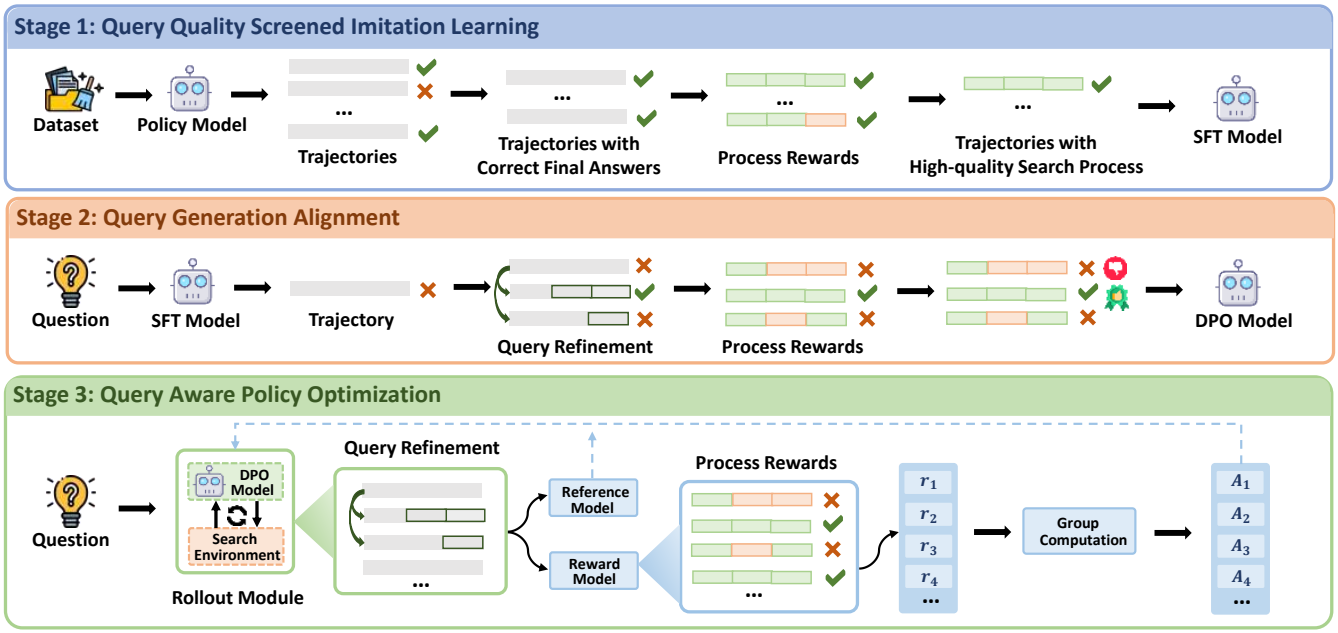


**Figure 3: The overall framework of query-oriented three-stage curriculum learning, including Query Quality Screened Imitation Learning, Query Generation Alignment, and Query Generation Alignment.**

practice, we begin by selecting trajectories that yield correct final answers and adhere to the proper format, thus guiding the model towards correct patterns from the outset. However, many trajectories, despite yielding correct final answers, contain low-quality intermediate search queries. Learning from such trajectories could lead the model to pick up suboptimal behaviors, thereby impairing its overall performance. To address this, we further leverage process rewards to selectively retain only those trajectories comprised entirely of high-quality intermediate search queries, i.e., $\forall t \in [0, \ldots, T], \mathcal{S}_t = 1$. This ensures that the trajectories comprising our final training dataset $\mathcal{D}$ not only yield correct final answers but also exhibit high-quality intermediate search queries. We then apply the standard SFT objective, which is formulated as:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(q,y)\sim\mathcal{D}} \left[\log P_\theta(y \mid q)\right], \tag{14}$$

where $q$ is the user's original query, $y$ is the agent's high-quality response, and $\theta$ denotes the model parameters.

***Stage-2: Query Generation Alignment.*** In this stage, the search agent cultivates advanced query generation capabilities through DPO training. Unlike common approaches that directly generate trajectories from scratch, we employ the query refinement mechanism when constructing comparative data. For each user's query $q$, the search agent first generates an initial trajectory $y_0$. Following this, each low-quality query within $y_0$ is refined and the search agent regenerates subsequent search steps from the refined query, producing a sequence of trajectories $y_1, \ldots, y_n$, where $n$ is the number of low-quality queries. This process ensures that for a given input $q$, the key differences among the candidate trajectories $y_0, y_1, \ldots, y_n$ originate specifically from the refined queries, thereby directly promoting the optimization of query generation.

Next, for each user's query $q$, we choose one positive sample $y_w$ and one negative sample $y_l$ among the corresponding candidate trajectories $y_0, y_1, \ldots, y_n$. Diverging from approaches that rely only on the correctness of the final answer, our selection criteria incorporate *both the final-answer correctness and the quality of intermediate search queries*, guided by the following principles:

- A trajectory with a correct final answer is preferred over one with an incorrect answer.
- Among trajectories with correct final answers, those with fewer low-quality (i.e., $\mathcal{S}_t = 0$) queries are preferred.
- Among trajectories with incorrect final answers, those containing more high-quality (i.e., $\mathcal{S}_t = 1$) queries are preferred.

We then optimize the model using the standard DPO objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(q,y_w,y_l)\sim\mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w \mid q)}{\pi_{\text{ref}}(y_w \mid q)} - \beta \log \frac{\pi_\theta(y_l \mid q)}{\pi_{\text{ref}}(y_l \mid q)} \right) \right]. \tag{15}$$

Here, $q$ is the user's original query, $y_w$ is the positive sample, $y_l$ is the negative sample, $\beta$ represents the hyperparameter, $\sigma$ refers to the sigmoid function, $\theta$ is the model parameters, and $\pi_{\text{ref}}$ indicates the reference model, which is initialized to $\pi_\theta$ and kept frozen throughout the entire training process.

***Stage-3: Query Aware Policy Optimization.*** In the final stage, we further enhance the search agent's integrated capabilities of information retrieval and utilization through Query Aware Policy Optimization. Specifically, we train it on a curated set of challenging questions that remained unresolved after multiple sampling trials. Unlike the standard GRPO algorithm that generates $G$ independent

trajectories from scratch, our method employs the query refinement mechanism as its rollout strategy. For each user's query, the search agent first generates an initial trajectory $y_0$ and then expands it into $y_0, y_1, \ldots, y_n$ through sequential refinement and regeneration. Different from the Query Generation Alignment stage, we retain at most $M$ trajectories from this set to avoid too many trajectories sharing a common prefix, thereby ensuring behavioral diversity and promoting the holistic improvement of the agent's capabilities. If the total number of trajectories collected remains less than $G$, we repeat this generation-and-expansion process, until a complete set of $G$ trajectories is obtained.

For reward design, we integrate process supervision into the reward function. Following Eq. (6), our reward function is:

$$r = r_{\text{composite}} + \lambda \cdot r_{\text{format}}, \tag{16}$$

where $\lambda$ is a weighting coefficient, $r_{\text{format}} \in \{0, 1\}$ indicates the correctness of the output format, and $r_{\text{composite}}$ defined in Eq. (7) integrates both outcome and process reward as follows:

$$r_{\text{composite}} = \begin{cases} \max(r_{\text{outcome}} - \gamma \cdot n_{\text{wrong}}, \phi_{\min}), & r_{\text{outcome}} = 1, \\ \min(r_{\text{outcome}} + \gamma \cdot n_{\text{correct}}, \phi_{\max}), & r_{\text{outcome}} = 0. \end{cases} \tag{17}$$

Here, $r_{\text{outcome}} \in \{0, 1\}$ denotes the final answer's correctness, $n_{\text{wrong}}$ and $n_{\text{correct}}$ represent the number of low- (i.e., $\mathcal{S}_t = 0$) and high-quality (i.e., $\mathcal{S}_t = 1$) queries respectively, $\gamma$ is a scaling factor for process rewards, and $\phi_{\min}, \phi_{\max}$ bound the influence of process rewards. This reward design incentivizes the agent not only to prioritize final answer correctness but also to refine its search process by reducing low-quality queries in successful trajectories. Moreover, even when unable to provide a final correct answer, the agent is motivated to generate more high-quality queries that may progressively approach the solution. We then optimize the model using the standard GRPO objective introduced in Section 3.2.

# 5 Experiments

## 5.1 Experimental Setup

***Dataset.*** To comprehensively evaluate the effectiveness of SmartSearch, we conduct experiments on two types of benchmarks: (1) knowledge-intensive tasks, including 2WikiMultihopQA [16], HotpotQA [59], Bamboogle [32], and Musique [46], which typically rely on local search over the Wikipedia corpus; and (2) web exploration tasks, including GAIA [31] and WebWalker [53], which usually require web search to gather information from online sources.

***Metrics.*** For a consistent comparison with previous studies, we use the widely adopted Exact March (EM) and word-level F1 score to evaluate the correctness of the final answer. To assess search efficiency, we follow prior work [5] and employ the Search Efficiency metric, defined as: $S_E = \frac{1}{N} \sum_{i=1}^{N} \frac{F_i}{T_i}$. Here, $N$ represents the total number of samples, $F_i$ denotes the F1 score for sample $i$, and $T_i$ represents the search call count for sample $i$. Additionally, to assess the quality of intermediate search queries, we introduce the Search Quality metric, defined as: $S_Q = \frac{1}{N} \left(C_{\text{perfect}} + C_{\text{partial}}\right)$ where $N$ represents the total number of samples, $C_{\text{perfect}}$ denotes the number of samples where the final answer is correct and all intermediate search queries are of high quality, and $C_{\text{partial}}$ denotes the number of samples where the final answer is incorrect but the trajectory contains high-quality intermediate search queries. In particular, we define the Perfect Rate as $\frac{1}{N} C_{\text{perfect}}$ and the Partial Rate as $\frac{1}{N} C_{\text{partial}}$, which jointly contribute to the overall Search Quality metric from two different aspects.

***Baselines.*** We compare SmartSearch against a set of representative baselines, which can be broadly categorized into three distinct categories: (1) prompt-based approaches, including Direct Inference, CoT [51], IRCoT [47], RAG [26], and Search-o1 [27]. (2) RL approaches with outcome rewards, including ReSearch [4], ZeroSearch [39], R1-Searcher [38], and Search-R1 [21]. (3) RL approaches with process rewards, including PPR [57], ReasonRag [65], and StepSearch [67].

***Implementation Details.*** Qwen2.5-3B-Instruct serves as the base model in SmartSearch and other baselines. Additionally, to improve efficiency, we train a smaller student model, Qwen2.5-3B-Instruct, to perform both scoring and query refinement tasks, with training labels annotated by the teacher model, Qwen3-32B. This teacher-student distillation strategy significantly reduces inference latency while maintaining competitive performance.

Our training is conducted under the LLaMA-Factory and VERL frameworks, using Asearcher-Base as the training dateset of our three-stage curriculum learning framework. In the process reward mechanism, we empirically set the hyperparameter $K$ to 1, which is used as a threshold to determine whether a query is novel based on the overlap count between the current retrieval results and those of previous queries.

In the Query Quality Screened Imitation Learning stage, we employ ARPO-14B [11] as the policy model for trajectory sampling. The trajectories obtained through this sampling process are then used to fine-tune the Qwen2.5-3B-Instruct model through SFT, resulting in the SFT model. Approximately 10k screened trajectories are selected for this SFT training. The training is conducted with a learning rate of 7e-6 over a total of 3 epochs, and the maximum input length during training is set to 16384 tokens. We utilize DeepSpeed ZeRO-3 [33] and FlashAttention2 [8] to accelerate training, with a total batch size of 64 and applying BF16 precision.

In the Query Generation Alignment stage, we perform DPO training using approximately 2k trajectories generated by the SFT model as positive and negative samples, resulting in the DPO model. This process involves LoRA fine-tuning with a learning rate of 7e-6 trained for 3 epochs, and the maximum input length during training is set to 10000 tokens. As in the previous stage, we leverage DeepSpeed ZeRO-3 and FlashAttention2 for efficient training, with a total batch size of 32 and BF16 precision.

In the Query Aware Policy Optimization stage, we focus on a curated set of challenging questions that remained unresolved after four sampling trials, with approximately 2k such trajectories selected for training. Through RL, the DPO model is further optimized to produce the final SmartSearch model. We set the hyperparameters $\phi_{\min} = 0.7$ and $\phi_{\max} = 0.3$, which bound the influence of process rewards in reward function. Additionally, we set $M = 2$, which limits the maximum number of retained trajectories sharing the same prefix. The training is conducted with a learning rate of 1e-6, where each sample undergoes 8 rollouts to explore different trajectories. The total training batch size is 64, with a PPO mini-batch size of 16. We set the maximum output length to 8192 tokens and limit the number of tool calls to 5 during each rollout.

**Table 1: Performance comparison of SmartSearch and existing approaches on four knowledge-intensive benchmarks, with bold for the best and underlined for the runner-up. Numbers in () indicate the improvement compared with the runner-up.**

| Method | 2WikiMQA | | HotpotQA | | Bamboogle | | Musique | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| ***Prompt-based Approaches*** | | | | | | | | | | |
| Direct Inference | 19.3 | 24.7 | 14.6 | 24.5 | 4.0 | 11.6 | 2.3 | 7.9 | 10.1 | 17.2 |
| CoT | 18.1 | 24.9 | 12.8 | 24.0 | 14.4 | 25.5 | 2.2 | 7.8 | 11.9 | 20.6 |
| IRCoT | 20.0 | 27.2 | 19.3 | 28.0 | 16.8 | 25.9 | 5.8 | 12.7 | 15.5 | 23.5 |
| RAG | 22.5 | 31.4 | 24.3 | 36.7 | 7.2 | 17.2 | 4.5 | 12.2 | 14.6 | 24.4 |
| Search-o1 | 20.9 | 29.4 | 22.0 | 33.6 | 28.8 | 36.1 | 5.1 | 12.6 | 19.2 | 27.9 |
| ***RL Approaches with Outcome Rewards*** | | | | | | | | | | |
| ReSearch | 29.4 | 36.7 | 28.5 | 40.8 | 12.8 | 22.9 | 10.0 | 17.3 | 20.2 | 29.4 |
| ZeroSearch | 29.2 | 36.5 | 27.5 | 39.1 | 14.4 | 25.4 | 10.4 | 18.2 | 20.4 | 29.8 |
| R1-Searcher | 29.8 | 37.1 | 27.0 | 38.7 | 31.2 | 39.2 | 8.0 | 16.4 | 24.0 | 32.9 |
| Search-R1 | 27.3 | 35.5 | 31.9 | 41.1 | 29.4 | 38.8 | 9.3 | 16.6 | 24.5 | 33.0 |
| ***RL Approaches with Process Rewards*** | | | | | | | | | | |
| ReasonRag | 36.5 | 43.2 | 32.2 | 41.7 | 30.4 | 39.1 | 11.3 | 18.6 | 27.6 | 35.7 |
| PPR | 33.7 | 41.8 | 38.1 | 50.3 | 31.2 | 39.4 | 14.7 | 22.0 | 29.4 | 38.4 |
| StepSearch | 32.1 | 38.9 | 35.1 | 45.9 | 36.8 | 48.4 | 16.6 | 24.9 | 30.1 | 39.5 |
| SmartSearch (Ours) | **45.3**(↑24%) | **52.3**(↑21%) | **40.7**(↑7%) | **52.4**(↑4%) | **44.8**(↑22%) | **56.1**(↑16%) | **19.1**(↑15%) | **27.8**(↑12%) | **37.5**(↑25%) | **47.2**(↑19%) |

**Table 2: Performance comparison of SmartSearch and baselines on web exploration tasks, with bold for the best.**

| Method | GAIA | | WebWalker | | Average | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| Search-o1 | 4.7 | 8.0 | 6.3 | 18.3 | 5.5 | 13.2 |
| Search-R1 | 6.3 | 9.8 | 7.5 | 21.6 | 6.9 | 15.7 |
| StepSearch | 9.4 | 12.5 | 9.1 | 25.8 | 9.3 | 19.2 |
| SmartSearch | **13.4** | **16.7** | **11.5** | **31.0** | **12.5** | **23.9** |

During the inference stage, we set the maximum output length per round to 8192 tokens and limit the total trajectory length to 16384 tokens, with a maximum of 10 tool calls per trajectory. For local search, we utilize the 2018 Wikipedia dump [24] provided by FlashRAG [22] as the knowledge corpus, and employ E5-base-v2 [49] as the retriever to obtain top 5 documents. For web search, the Serper API is employed to retrieve the top 10 document snippets.

## 5.2 Main Result

***Overall Performance.*** Table 1 presents the main results, demonstrating that SmartSearch consistently surpasses existing approaches across four datasets, and yielding several important insights.

**(1) Prompt-based approaches exhibit limited performance.** Direct Inference and CoT, which are based entirely on the model's internal knowledge, achieve an average EM of only around 10%, highlighting the inherent challenges of LLMs, including hallucinations and static parametric knowledge. RAG and IRCoT yield a gain of around 5% in average EM, demonstrating the necessity of integrating external knowledge. Among prompt-based methods, Search-o1 attains the highest performance, reaching an average EM score of 19.2%, reflecting the effectiveness of search agents. However, it still lags behind other fine-tuning-based approaches.

**(2) Incorporating process rewards effectively enhances RL training.** While outcome-driven RL methods such as ReSearch, Search-R1, and R1-Searcher improve performance over prompt-based approaches, indicating the benefits of RL for LLM-based search agents, they often remain inferior to RL approaches that integrate both outcome and process rewards by a margin of around 5% in both average EM and F1 score. This underscores that reward signals based solely on final outcomes result in sparse feedback. Such sparse feedback provides insufficient guidance for intermediate steps and leads to unstable optimization, thereby highlighting the critical importance of fine-grained supervision.

**(3) Optimizing the quality of intermediate search queries significantly improves overall performance.** Existing methods often overlook the quality of intermediate search queries, which can lead to stagnation in information retrieval abilities. By explicitly optimizing the quality of intermediate search queries under the guidance of process rewards, SmartSearch enhances the search agent's overall effectiveness, achieving more than 7% improvement in both average EM and F1 score compared with other process-supervised RL methods.

***Generalization to Web Search Scenarios.*** As discussed earlier, SmartSearch is trained solely on Wikipedia-based local search. To evaluate its generalization ability to web search, we test it against several baseline models on two demanding web exploration tasks, GAIA and WebWalker. As demonstrated in Table 2, SmartSearch surpasses existing approaches across both datasets, achieving an average F1 score increase of nearly 5%. This indicates that while SmartSearch optimizes the quality of intermediate search queries in the local search setting, it also exhibits strong generalization

**Table 3: Ablation study results for the two core mechanisms in SmartSearch across all three stages of curriculum learning training framework, with bold for the best results of each stage.**

| Method | 2WikiMQA | | HotpotQA | | Bamboogle | | Musique | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| ***Stage 1*** | | | | | | | | | | |
| SmartSearch | **38.2** | **45.3** | **35.3** | **45.5** | **38.4** | **51.0** | **14.7** | **21.6** | **31.7** | **40.9** |
| w/o process rewards | 33.8 | 40.6 | 32.5 | 42.8 | 36.0 | 48.7 | 12.6 | 20.8 | 28.7 | 38.2 |
| ***Stage 2*** | | | | | | | | | | |
| SmartSearch | **41.4** | **48.7** | **37.9** | **48.5** | **39.2** | **51.8** | **15.4** | **23.6** | **33.5** | **43.2** |
| w/o process rewards | 40.2 | 47.4 | 36.5 | 47.2 | 37.6 | 50.1 | 14.4 | 22.3 | 32.2 | 41.8 |
| w/o query refinement | 39.2 | 46.7 | 35.6 | 46.1 | 36.0 | 49.6 | 14.6 | 22.9 | 31.4 | 41.3 |
| ***Stage 3*** | | | | | | | | | | |
| SmartSearch | **45.3** | **52.3** | **40.7** | **52.4** | **44.8** | **56.1** | **19.1** | **27.8** | **37.5** | **47.2** |
| w/o process rewards | 43.3 | 50.6 | 40.0 | 51.5 | 39.2 | 51.6 | 17.9 | 26.7 | 35.1 | 45.1 |
| w/o query refinement | 44.1 | 51.2 | 39.9 | 51.6 | 41.6 | 54.2 | 17.5 | 26.4 | 35.8 | 45.9 |
| Standard GRPO | 43.6 | 50.8 | 39.0 | 50.6 | 40.8 | 52.5 | 15.9 | 24.8 | 34.8 | 44.7 |

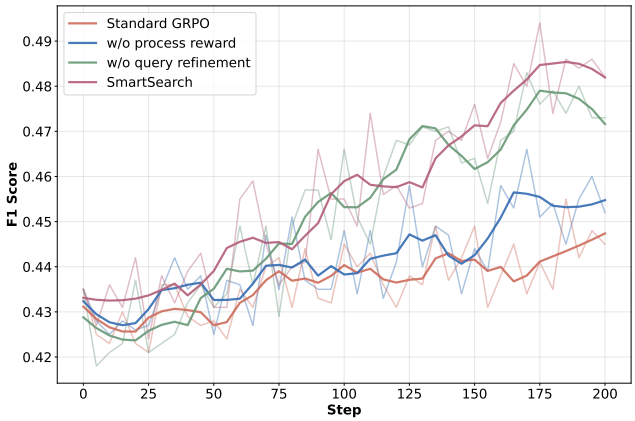


**Figure 4: F1 score training dynamics for different algorithms.**

capabilities in the web search environment, maintaining robust performance despite the difference in settings.

## 5.3 Ablation Study

To further examine the impact of SmartSearch's two key mechanisms—process rewards and query refinement, we conduct extensive ablation studies across all three training stages. The results are summarized in Table 3.

For Stage 1, we compare our configuration with a baseline that filters the training data exclusively according to whether the final answer is correct. The results indicate that incorporating query-quality filtering enables the model to achieve superior performance with only 60% of the training data, highlighting the importance of learning from trajectories with high-quality search processes.

For Stage 2, we compare our method with two alternatives: (1) directly generating full trajectories without query refinement, and (2) determining preference based exclusively on the final answer correctness. Ablation results underscore the essential contribution of both mechanisms in this stage, particularly the query refinement mechanism, underscoring the significance of promoting the optimization of query generation.

For Stage 3, we compare our algorithm with three variants: a GRPO baseline, a version that only incorporates the process rewards into the reward function, and a version that only applies the query refinement mechanism during rollout. As depicted in Figure 4, we demonstrate F1 score curves of various RL algorithms during training. The results demonstrate that our algorithm reliably outperforms all alternatives. Notably, integrating process rewards into the reward function consistently yields significant gains, illustrating the crucial role of fine-grained supervision for the quality of each intermediate search query.

## 5.4 Quantitative Analyses

To comprehensively assess the effectiveness of the SmartSearch framework, we perform multiple quantitative experiments. The following analyses demonstrate its superiority in four key aspects: intermediate query quality, search efficiency, the effectiveness of the process reward model, and the effectiveness-efficiency trade-off.

***Search Query Quality Analysis.*** To assess whether SmartSearch improves the quality of intermediate search queries, we compare the Search Quality metric across various methods. As presented in Figure 5 (a), SmartSearch achieves the highest Search Quality, with the highest values for both Perfect Rate and Partial Rate, which collectively contribute to the overall Search Quality metric. This indicates that SmartSearch effectively enhances the quality of intermediate search queries. Specifically, the search agent reduces ineffective searches while striving to generate perfect trajectories where all queries are of high quality. Even when unable to provide a final correct answer, the agent makes more attempts to generate high-quality queries that edge incrementally closer to the correct solution.

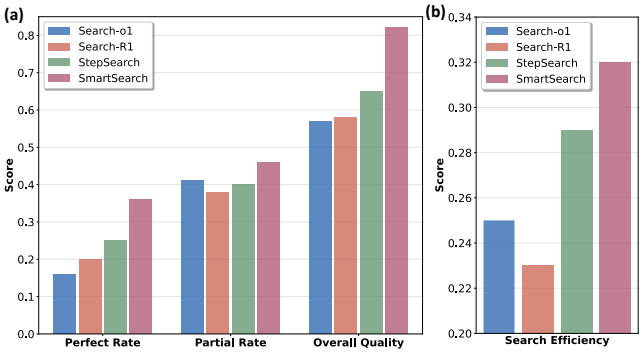


**Figure 5: Left: Search query quality comparison. Right: Search efficiency comparison.**

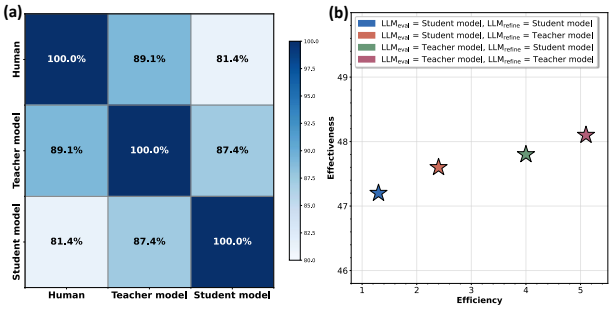


**Figure 6: Left: Overlap between the scores assigned to intermediate search queries by the student model, the teacher model, and human annotations. Right: Effectiveness and efficiency tradeoff in SmartSearch.**

***Search Efficiency Analysis.*** The results in previous sections have clearly demonstrated that SmartSearch outperforms all baselines in terms of answer accuracy across various benchmarks. We now further evaluate whether it also achieves superior search efficiency, which is a critical factor for practical deployment. To this end, we compare the search efficiency metrics across multiple methods, and as shown in Figure 5 (b), SmartSearch outperforms all other methods in this regard. This suggests that by effectively optimizing the quality of intermediate search queries, SmartSearch is able to generate more precise and targeted queries, thereby significantly reducing ineffective or failed search rounds and, as a result, improving overall search efficiency.

***Process Reward Model Analysis.*** The process reward model plays a crucial role in our approach by providing fine-grained supervision for the quality of each intermediate search query and guiding subsequent query refinement. To assess the effectiveness of our process reward model, we randomly choose 100 trajectories generated by SmartSearch on the test set. For each intermediate search query in these trajectories, we compare the scores annotated by three sources: the teacher model, the student model, and human annotators. Figure 6 (a) illustrates the overlap between the scores assigned to each intermediate search query by these three sources. The results reveal that the teacher model achieves nearly 90% overlap with human annotations, demonstrating its effectiveness in labeling the training data. After training, the student model achieves over 85% overlap with the teacher model, indicating that the distillation process successfully transfers the scoring capability from the teacher to the student. Finally, the student model shows over 80% overlap with human annotations, a result that is entirely acceptable for practical deployment. This striking balance between scoring accuracy and computational efficiency makes the student model a suitable choice for our framework.

***Effectiveness-Efficiency Trade-off.*** To validate the suitability of the lightweight student model as both $\text{LLM}_{\text{eval}}$ and $\text{LLM}_{\text{refine}}$, we conduct experiments using a more powerful teacher model in these roles. In this experiment, effectiveness is defined as the average F1 score, while efficiency refers to the average time (in seconds) required to apply the process rewards and query refinement mechanisms to each sample. As shown in Figure 6 (b), using a more powerful model as both $\text{LLM}_{\text{eval}}$ and $\text{LLM}_{\text{refine}}$ indeed improves the agent's effectiveness, underscoring the importance of the process rewards and query refinement mechanisms within our training framework. However, this gain in average F1 score is relatively modest, remaining below 1% across all settings. In contrast, the time required to process each sample increases by nearly five times. This result demonstrates a clear trade-off between effectiveness and efficiency, and the decision to use the lightweight student model as both $\text{LLM}_{\text{eval}}$ and $\text{LLM}_{\text{refine}}$ proves to be a sensible one. This choice strikes an ideal optimal balance between effectiveness and efficiency, thereby ensuring effective query optimization while avoiding excessive computational costs.

## 6 Conclusion

In this work, we introduce SmartSearch, a novel framework designed to optimize the quality of intermediate search queries through two key mechanisms: (1) Process rewards, which provide fine-grained supervision for the quality of each intermediate search query through Dual-Level Assessment. (2) Query refinement, which promotes the optimization of query generation by selectively refining low-quality queries and regenerating subsequent search rounds from these refined points. Building on the foundation of the two mechanisms, we design a three-stage curriculum learning framework that guides the agent through a progression from imitation and alignment to generalization, enabling it to progressively internalize the ability to enhance query quality. Experiments across fix challenging benchmarks demonstrate that SmartSearch consistently surpasses existing baselines, with further rigorous quantitative analyses confirming significant gain in both search efficiency and query quality.

## Acknowledgments

This work was supported by National Natural Science Foundation of China No. 62272467 and No. 625B2178. The work was partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

## References


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

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

[3] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17754–17762.

[4] Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, et al. 2025. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470* (2025).

[5] Yifei Chen, Guanting Dong, and Zhicheng Dou. 2026. Toward Effective Tool-Integrated Reasoning via Self-Evolved Preference Learning. In *14th International Conference on Learning Representations, ICLR 2026*.

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

[7] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. https://proceedings.mlr.press/v267/chu25c.html

[8] Tri Dao. 2024. FLASHATTENTION-2: FASTER ATTENTION WITH BETTER PARALLELISM AND WORK PARTITIONING. In *12th International Conference on Learning Representations, ICLR 2024*.

[9] Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, et al. 2025. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800* (2025).

[10] Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2026. Tool-Star: Empowering LLM-Brained Multi-Tool Reasoner via Reinforcement Learning. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[11] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849* (2025).

[12] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Ji-Rong Wen, and Zhicheng Dou. 2025. Understand what LLM needs: Dual preference alignment for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*. 4206–4225.

[13] Tianqing Fang, Zhisong Zhang, Xiaoyang Wang, Rui Wang, Can Qin, Yuxuan Wan, Jun-Yu Ma, Ce Zhang, Jiaqi Chen, Xiyun Li, et al. 2025. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training. *arXiv preprint arXiv:2508.00414* (2025).

[14] Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond Ten Turns: Unlocking Long-Horizon Agentic Search with Large-Scale Asynchronous RL. In *First Workshop on Multi-Turn Interactions in Large Language Models*.

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

[16] 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*. 6609–6625.

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

[18] Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2025. Rag-star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 7064–7074.

[19] Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. DeepRetrieval: Hacking Real Search Engines and Retrievers with Large Language Models via Reinforcement Learning. *CoRR* (2025).

[20] Yi Jiang, Lei Shen, Lujie Niu, Sendong Zhao, Wenbo Su, and Bo Zheng. 2025. QAgent: A modular Search Agent with Interactive Query Understanding. *arXiv preprint arXiv:2510.08383* (2025).

[21] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. *CoRR* abs/2503.09516 (2025). arXiv:2503.09516 doi:10.48550/ARXIV.2503.09516

[22] Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*. 737–740.

[23] Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5591–5606.

[24] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering.. In *EMNLP (1)*. 6769–6781.

[25] Yongqi Leng, Yikun Lei, Xikai Liu, Meizhi Zhong, Bojian Xiong, Yurong Zhang, Yan Gao, Yao Hu, Deyi Xiong, et al. 2025. DecEx-RAG: Boosting Agentic Retrieval-Augmented Generation with Decision and Execution Optimization via Process Supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 1412–1425.

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

[27] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*. 5420–5438. doi:10.18653/V1/2025.EMNLP-MAIN.276

[28] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[29] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. Retrollm: Empowering large language models to retrieve fine-grained evidence within generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 16754–16779.

[30] Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. 2025. OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

[31] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for General AI Assistants. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. https://openreview.net/forum?id=fibxvahvs3

[32] 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*. 5687–5711.

[33] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3505–3506.

[34] Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922* (2023).

[35] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551.

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

[37] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. 2023. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617* (2023).

[38] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning. *CoRR* (2025).

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

[40] Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G Nestor, Ali Soroush, Pierre A Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F Rousseau, et al. 2023. Evaluating large language models on medical evidence summarization. *NPJ digital medicine* 6, 1 (2023), 158.

[41] Zhengwei Tao, Haiyang Shen, Baixuan Li, Wenbiao Yin, Jialong Wu, Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Liwen Zhang, et al. 2025. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. *arXiv preprint arXiv:2510.24697* (2025).

[42] Kimi Team. 2025. Kimi K2: Open Agentic Intelligence. *CoRR* abs/2507.20534 (2025). arXiv:2507.20534 doi:10.48550/ARXIV.2507.20534

[43] Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face* (2024).

[44] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[46] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* 10 (2022), 539–554.

[47] 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 (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. 10014–10037. doi:10.18653/V1/2023.ACL-LONG.557

[48] Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. 2025. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn LLM Agents. *arXiv preprint arXiv:2510.14967* (2025).

[49] 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* (2022).

[50] Yiding Wang, Zhepei Wei, Xinyu Zhu, and Yu Meng. 2025. Beyond Outcome Reward: Decoupling Search and Answering Improves LLM Agents. *arXiv preprint arXiv:2510.04695* (2025).

[51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[52] Tongyu Wen, Chenglong Wang, Xiyuan Yang, Haoyu Tang, Yueqi Xie, Lingjuan Lyu, Zhicheng Dou, and Fangzhao Wu. 2025. Defending against Indirect Prompt Injection by Instruction Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*. 19472–19487. https://aclanthology.org/2025.findings-emnlp.1060/

[53] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. WebWalker: Benchmarking LLMs in Web Traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. 10290–10305. https://aclanthology.org/2025.acl-long.508/

[54] Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. 2025. HiPRAG: Hierarchical Process Rewards for Efficient Agentic Retrieval Augmented Generation. *arXiv preprint arXiv:2510.07794* (2025).

[55] Guangzhi Xiong, Qiao Jin, Xiao Wang, Yin Fang, Haolin Liu, Yifan Yang, Fangyuan Chen, Zhixing Song, Dengyu Wang, Minjia Zhang, et al. 2025. Rag-gym: Optimizing reasoning and search agents with process supervision. *arXiv preprint arXiv:2502.13957* (2025).

[56] Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. https://openreview.net/forum?id=farT6XXntP

[57] Peiran Xu, Zhuohao Li, Xiaoying Xing, Guannan Zhang, Debiao Li, and Kunyu Shi. 2025. Hybrid Reward Normalization for Process-supervised Non-verifiable Agentic Tasks. *arXiv preprint arXiv:2509.25598* (2025).

[58] Zhichao Xu, Zongyu Wu, Yun Zhou, Aosong Feng, Kang Zhou, Sangmin Woo, Kiran Ramnath, Yijun Tian, Xuan Qi, Weikang Qiu, et al. 2025. Beyond Correctness: Rewarding Faithful Reasoning in Retrieval-Augmented Generation. *arXiv preprint arXiv:2510.13272* (2025).

[59] 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*. 2369–2380.

[60] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

[61] Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*.

[62] Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*. PMLR, 41092–41110.

[63] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. 2026. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. *Trans. Mach. Learn. Res.* 2026 (2026). https://openreview.net/forum?id=RY19y2RI1O

[64] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics* 12 (2024), 39–57.

[65] Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, et al. 2025. Process vs. Outcome Reward: Which is Better for Agentic RAG Reinforcement Learning. *arXiv preprint arXiv:2505.14069* (2025).

[66] Qingfei Zhao, Ruobing Wang, Dingling Xu, Daren Zha, and Limin Liu. 2025. R-Search: Empowering LLM Reasoning with Search via Multi-Reward Reinforcement Learning. *arXiv preprint arXiv:2506.04185* (2025).

[67] 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, EMNLP 2025, Suzhou, China, November 4-9, 2025*. 21805–21830. doi:10.18653/V1/2025.EMNLP-MAIN.1106