StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization

Anonymous ACL submission

Abstract

Efficient multi-hop reasoning requires Large Language Models (LLMs) based agents to acquire high-value external knowledge iteratively. Previous work has explored reinforcement learning (RL) to train LLMs to perform search-based document retrieval, achieving notable improvements in QA performance, but underperform on complex, multi-hop QA resulting from the *sparse rewards from global signal only*. To address this gap in existing research, we introduce **StepSearch**, a framework for search LLMs that trained with *step-wise* proximal policy optimization method. It consists of richer and more detailed intermediate search rewards and token-level process supervision based on information gain and redundancy penalties to better guide each search step. We constructed a fine-grained question-answering dataset containing sub-question-level search trajectories based on open source datasets through a set of data pipeline method. On standard multi-hop QA benchmarks, it significantly outperforms global-reward baselines, achieving **11.2%** and **4.2%** absolute improvements for 3B and 7B models over various search with RL baselines using only 19k training data, demonstrating the effectiveness of fine-grained, stepwise supervision in optimizing deep search LLMs.

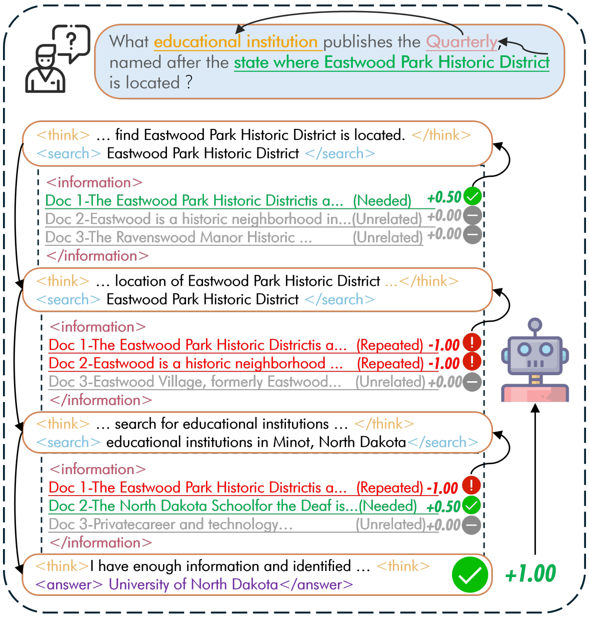


Figure 1: Step-wise search involves interactive rounds, with information gain being rewarded and redundancy penalised. Each interaction evaluates thinking and searching behaviour based on the retrieved results, with the final answer being used as the basis for global rewards.

1 Introduction

Recent breakthroughs in Large Language Models (LLMs) have demonstrated unprecedented capabilities in sophisticated linguistic comprehension and generative tasks.

Reinforcement learning enhanced architectures(*e.g.*, OpenAI-o3 (Jaech et al., 2024), DeepSeek-R1 (DeepSeek-AI et al., 2025), and Kimi-1.5 (Team et al., 2025)) employ policy-gradient methods (PPO (Schulman et al., 2017), GRPO (Shao et al., 2024)) to advance multi-hop logical reasoning (Xie et al., 2025). However, complex multi-hop QA still suffers from intrinsic knowledge gaps (Lee and Roh, 2024) and static, inefficient knowledge-assimilation mechanisms (Jin et al., 2024; Schick et al., 2023). To address limited modeling of internal dependencies, recent work has adopted prompting strategies, RAG architectures, and tailored learning paradigms (Patil, 2025; Lewis et al., 2020).

Chain-of-Thought (CoT) prompting (Wei et al., 2022) decomposes complex inference into sequential subtasks but remains highly sensitive to prompt formulation and does not eliminate hallucinations. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) dynamically incorporates external corpora to bridge knowledge gaps and suppress spurious content (Zhao et al., 2024; Gupta et al., 2024; Fan et al., 2024); embedding structured knowledge graphs further enforces semantic coherence through explicit entity relations (Soman et al., 2024; Edge et al., 2025). Agentic frameworks—combining self-reflection, strategic planning, and multi-agent collaboration—facilitate adaptive task decomposition and iterative refinement (Singh et al., 2025; Li et al., 2025). Advanced retrieval tactics (query reformulation, reranking, hybrid vector–keyword indexing) bolster multi-hop reasoning while filtering noise (Glass et al., 2022; Sawarkar et al., 2024). Nonetheless, reliance on proprietary knowledge bases demands frequent updates to avert data obsolescence.

Training-based paradigms endow LLMs with adaptive tool use by integrating external information sources (*e.g.*, search engines) directly into the training loop. Supervised fine-tuning (SFT) frameworks—such as ToolFormer (Schick et al., 2023), ToolKengPT (Hao et al., 2023), and related efforts (Qu et al., 2025; Shi et al., 2025)—substantially boost performance in specialized, knowledge-intensive tasks but suffer from poor out-of-domain generalization (Chu et al., 2025).

Recent advances have adopted reinforcement learning to learn dynamic retrieval policies, enabling models to iteratively query and integrate external knowledge based on the static RAG paradigm (Huang et al., 2025; Jiang et al., 2025). Methods such as R1-Searcher (Jin et al., 2025), Search-R1 (Song et al., 2025), ReSearch (Chen et al., 2025) and ZeroSearch (Sun et al., 2025) rely on answer and format-level rewards, empower agents to autonomously invoke search tools and achieve QA performance surpassing conventional RAG. DeepResearcher (Zheng et al., 2025) further extends this paradigm to unconstrained online search environments, highlighting the scalability and potential of search-RL approaches. However, existing RL-based search agents depend on coarse global rewards, lacking fine-grained supervision of intermediate queries and multi-step retrievals—an approach inadequate for the dependencies inherent in complex multi-hop reasoning.

Process-level supervision enables the design of fine-grained reward functions that steer strategic query planning and enhance retrieval quality in complex search environments (Zhu et al., 2025; Ye et al., 2025b,a; Wang et al., 2025). However, existing step-reward methods—such as R1-VL (Zhang et al., 2025) for pure logical reasoning and RPO (Liu et al., 2024) lack true token-level supervision for interactive retrieval tasks. Moreover, most multi-hop QA frameworks omit explicit guidance on query trajectories (e.g., intermediate search keywords or document usage), leaving a critical gap in search-path modeling.

To address these shortcomings, We propose **StepSearch**, a reinforcement learning framework that integrates iterative retrieval with explicit **step-wise** supervision for search llms (Figure 1). Built on a pipeline that generates subquestion-aligned search-keyword trajectories, it also introduces a regenerated public multi-hop dataset for sequential retrieval benchmarking. By augmenting PPO with token-level rewards that combine information gain and redundancy penalties, StepSearch boosts policy convergence and improves retrieval fidelity and QA accuracy.

In general, our core contribution lies in:

- **Universal multi-hop search data.** We develop a novel MuSiQue-based pipeline, contributing 60k filtered sub-question search keywords that generalize across retrieval datasets.
- **StepSearch: Step-wise RL with dual rewards.** We augment PPO with token-level rewards—information gain and redundancy penalties—for both query formulation and document retrieval.
- **State-of-the-art performance.** StepSearch outperforms standard RL baselines by **5.7%**, **9.1%**, **10.0%**, and **15.2%** absolutely on diverse multi-hop QA benchmarks.

2 Methodology

2.1 Data Augmentation Pipeline

In this pilot study, we construct a multi-turn Q&A dataset with subquestion-level search trajectories. Starting from the MusiQue (Trivedi et al., 2022) dataset, our pipeline show as (Figure. 2):

(a) Leverage `GPT-4o`. to enrich decomposed MuSiQue questions with coherent subquestion–answer pairs, then derive N search queries per step for retrieval.

(b) Each enhanced step question is then reformulated into a set of N search queries to facilitate information retrieval.

(c) Queries are issued to M sources (*e.g.*, Google, Bing, Wiki-18), and only those returning valid results in at least $\lceil M/2 \rceil$ sources are retained.

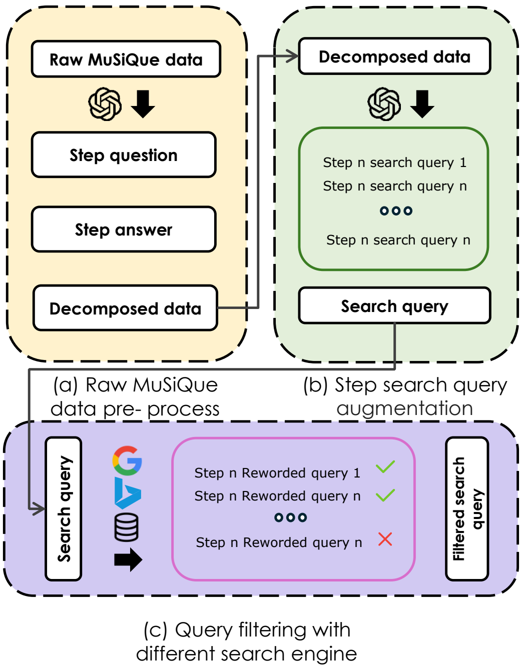


Figure 2: Data pipeline for generating the corresponding search query for the Q&A intermediate process.

2.2 Train LLM with Search Actions

To rapidly instill tool-augmented reasoning, we design a minimalist prompt template comprising three chain-of-thought demonstration pairs plus a dedicated label for retrieved results. Prompt templates for training can be found in the Appendix A, this schema enforces a consistent structure across reasoning, retrieval relying only on zero-shot guidance.

Rollout and **Mask** for **Retrieve: (1)** We run the loop of <think>...</think>, <search>...</search>, <information>...</information> iteratively, appending external docs until LLM returns <answer>...</answer> or the action budget is reached. **(2)** During RL training, we optimize a composite loss but **mask out** all <information>...</information> segments from gradient computation, thereby decoupling parameter updates from retrieval artifacts and focusing learning on the model's internal reasoning and search-policy parameters, as established in prior search-RL work (Jin et al., 2025; Song et al., 2025; Chen et al., 2025; Sun et al., 2025; Zheng et al., 2025).

2.3 StepSearch

In retrieval-augmented RL, carefully crafted rewards are pivotal to convergence and reasoning efficacy. In addition to the standard format and final-answer reward r_{answer}, we introduce a **search-key reward** r_{key} to promote informative query issuance directly. Our method further diverges from vanilla PPO by segmenting each turn into **think** $\rightarrow$ **search** $\rightarrow$ **answer** phases and assigning **token-level rewards**: each token earns an information-gain signal $\mathcal{G}^t$ and incurs a redundancy penalty $\mathcal{P}^t$. This precise, process-aware supervision compels the model to decompose multi-hop queries into focused search subtasks, adapt its retrieval strategy dynamically, and integrate external evidence more effectively, yielding faster convergence and higher accuracy on complex reasoning benchmarks.

Our optimization algorithm combines the above-mentioned Search Steps supervision reward based on the currently widely used actor-critic approach PPO (Schulman et al., 2017), denote as **StePPO**. For each sample input $x \backsim \mathcal{D}$, obtain output o from the old policy $\pi_{\theta_{old}}$, let $I(y_t)$ be the token-loss masking indicator, it equals 1 when o_t is generated by actor LLM else 0 for retrieved tokens. Then optimize the policy π_θ with the reference policy $\pi_{\theta_{ref}}$ by maximizing the following objective:

$$
\begin{aligned}
&\mathcal{J}_{\text{StePPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, o\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \\
&\left\{ \frac{1}{\sum_{t=1}^{|o|} I(o_t)} \sum_{t=1:I(o_t)=1}^{|o|} \left[\frac{\pi_\theta(o_t|x, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|x, o_{<t})} A_t, \right.\right. \\
&\left.\left. \text{clip}\left(\frac{\pi_\theta(o_t|x, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|x, o_{<t})}, 1-\epsilon, 1+\epsilon \right) A_t \right] \right\},
\end{aligned} \tag{1}
$$

here, ϵ is a hyper-parameter for clipping to stalblilize training, and A_t represents the estimated advantage computed with GAE algorithm (Schulman et al., 2015), based on future rewards $r_{\geq t}$, which is composed of the gloabal and step-wise search round rewards, and a learned value function V_ϕ. The global reward is set at the last position of the output, while the step-wise reward is set at the last token of each round of search behavior.

2.3.1 Type 1 Reward: Global Signal

Format Requirement: To ensure the model adopts the prescribed multi-step "*search + reason*" workflow and correctly initiates search actions across iterative reasoning rounds, we enforce strict format validation as a hard constraint rather than implement it as an explicit reward. The required output format is defined as follows:

- Only the search queries in the proper <search>...</search> pairs will be extracted

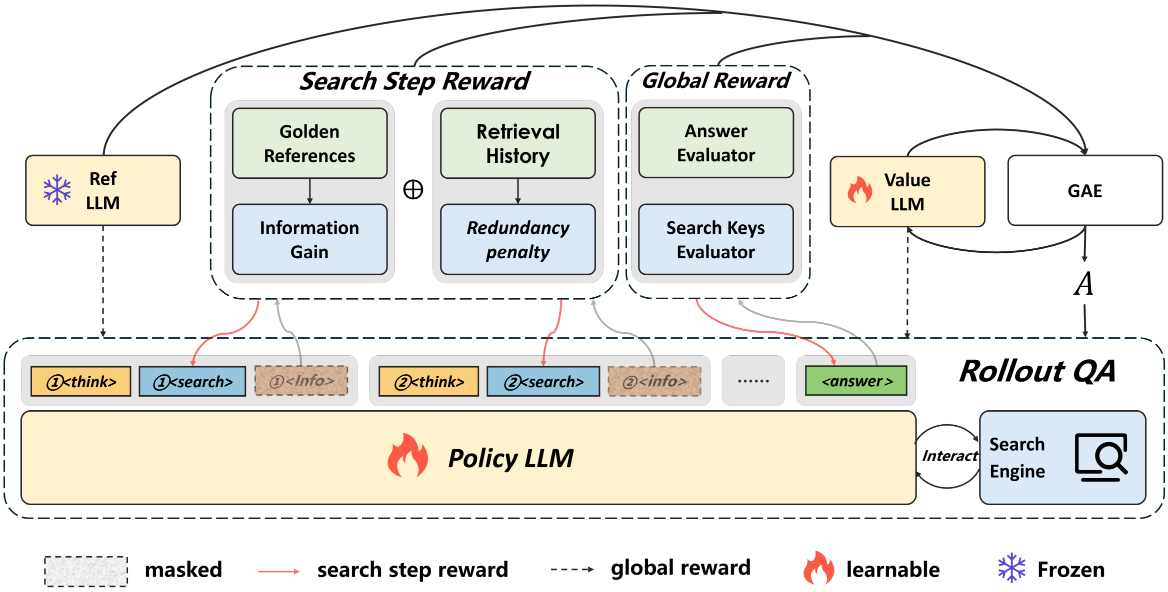


Figure 3: Overview of StepSearch. At each step, the model issues queries to an external engine and receives snippets. **Search Step Reward** score, combining information gain and redundancy penalty, are applied to tokens within each round, while the **global reward**, based on final answer accuracy and keyword hit rate, is applied at the last token. Retrieved content is masked during training to isolate the model's generative parameters.

and used to call search tools, and the answer must be in the <answer>...</answer> pair.

- At least one round of "think" and "search" behaviour
- Only one <answer>...</answer> tag pair to answer the question, and it must be at the end.

Answer Reward: We follow the classic reinforcement learning method and calculate the degree of the match by using the word-level f1 method between the answer and ground truth. Let PN represent the word count of the predicted answer, RN for word count of the golden answer and IN stands for the word count of the intersection between them, then the answer reward r_{answer} can be defined as:

$$\mathrm{F1}(x, y) = \frac{2 * IN}{PN + RN} \tag{2}$$

$$r_{\text{answer}} = \begin{cases} \mathrm{F1}(a_{\text{pred}}, a_{\text{gt}}), & \text{format is correct,} \\ 0, & \text{format is incorrect.} \end{cases} \tag{3}$$

Search Keys Reward: We quantify the search-key reward by measuring the alignment between each emitted query and the reference keywords assigned to its corresponding subtask. Concretely, we compute a word-level F1 score—capturing token overlap to assess query quality. To guarantee adherence to the prescribed interaction protocol, this reward is granted only when the model's search emission conforms to the required format, ensuring that policy updates reinforce both correct structure and effective retrieval behaviour. Suppose there are T rounds of queries $Q = \{q_1, q_2, \ldots, q_T\}$, $K_i = \{k_{i1}, k_{i2}, \ldots, k_{iN_i}\}, i = 1, \ldots, M$ corresponding M subquestions which each contains N_i related golden queries, thus, the search keyword reward can be calculated as:

$$f_{ijt} = \mathrm{F1}(q_t, k_{ij}), \tag{4}$$

$$\begin{aligned} r_{\text{key}} &= \frac{1}{M} \sum_{i=1}^{M} \left(\max_{1 \le j \le N_i} \left(\max_{1 \le t \le T} f_{ijt} \right) \right) \\ &= \frac{1}{M} \sum_{i=1}^{M} \max_{1 \le j \le N_i} \max_{1 \le t \le T} f_{ijt}. \end{aligned} \tag{5}$$

Type 1 Reward: Set γ_{key} as the scale factor and the final reward rule can be expressed by the following formula:

$$r_{\text{overall}} = r_{\text{answer}} + \gamma_{\text{key}} \cdot r_{\text{key}}. \tag{6}$$

2.3.2 Type 2 Reward: Search Step

The step-wise reward r^t_{step} of each round of search behaviour can be expressed as information gain $\mathcal{G}^t$

minus redundancy penalty $\mathcal{P}^t$ as **Type 2 Reward:**

$$r^t_{\text{step}} = \mathcal{G}^t - \mathcal{P}^t. \tag{7}$$

Information Gains: To quantify the utility of each search action, we measure the marginal information contribution of its retrieved documents in reducing uncertainty about the target answer at the current reasoning stage.

Let $D^g = \{d^g_1, \ldots, d^g_n\}$ denote the n ground-truth documents required to resolve problem p at search turn t, where each d^g_i contains the gold-standard information for a specific subtask. We maintain a memory vector $M^t = [m^t_1, \ldots, m^t_n]$, in which m^t_i records the maximum similarity observed to date between any retrieved document and d^g_i. At turn t, the agent retrieves a set $D^{r(t)} = \{d^{r(t)}_1, \ldots, d^{r(t)}_k\}$ of k documents; we denote by c^t_j the similarity between $d^{r(t)}_j$ and its corresponding golden document(s). To evaluate this alignment, we adopt a submodular coverage function instantiated with cosine similarity over *TF–IDF* (Ramos et al., 2003) representations, which naturally enforces diminishing returns and penalizes redundant retrievals.

First initialize m^t_i to 0, the current matching degree of the round t search results can be calculated based on each golden info document, and the highest similarity among the search documents in this round is taken as c^t_i:

$$c^t_i = \max_{1 \le j \le k} \frac{\overrightarrow{d^g_i} \cdot \overrightarrow{d^{r(t)}_j}}{||\overrightarrow{d^g_i}|| \cdot ||\overrightarrow{d^{r(t)}_j}||}, \quad i = 1, \cdots, n. \tag{8}$$

The valuable information gain $\triangle^t_i$ on golden document d^g_i of this round t is calculated based on the current matching degree c^t_i of this round and the global maximum matching degree m^t_i of the previous round:

$$\triangle^t_i = \max(c^t_i - m^t_i, 0), \quad i = 1, \cdots, n, \tag{9}$$

then, the overall information gain value of tth round is the average gain of n golden info documents in the current round:

$$\mathcal{G}^t = \frac{1}{n}\sum_{i=1}^{n} \triangle^t_i = \frac{1}{n}\sum_{i=1}^{n} \max(c^t_i - m^t_i, 0). \tag{10}$$

Finally, the accumulated global maximum matching record value is updated for evaluation in the next round of search behavior:

$$m^t_i = max(m^{t-1}_i, c^t_i), \quad i = 1, \cdots, n. \tag{11}$$

Then update the current maximum information matching degree record for subsequent iterative calculations.

Redundancy Penalty: During search-stage supervision, we observed that repetitive confirmatory queries both waste budget and amplify hallucinations without effective feedback. To counter this, we track a cumulative retrieval history H^t (with $H^0 = \varnothing$) and let each round's retrieved set be I^t. Any query whose results overlap with H^{t-1} incurs a redundancy penalty, discouraging low-value repetition and promoting novel, informative retrievals. At the end of round t, we update

$$H^t = H^{t-1} \cup I^t, \tag{12}$$

the redundancy penalty value $\mathcal{P}^t$ of the tth round can be expressed by counting the proportion of the documents retrieved $D^{r(t)}$ in this round that are repeated in any previous round:

$$\mathcal{P}^t = \frac{1}{k}\sum_{j=1}^{k} 1(d^{r(t)}_j \in H^{t-1}), \tag{13}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

3 Experiment

3.1 Dataset and Evaluation Metrics

During training with process supervision, Our empirical benchmarks span four established multi-hop Q&A datasets: (1) **HotpotQA** (Yang et al., 2018), (2) **MuSiQue** (Trivedi et al., 2022), (3) **2WikiMultiHopQA** (Ho et al., 2020), and (4) **Bamboogle** (Press et al., 2022).

To maintain alignment with prior work (Yu et al., 2024; Jin et al., 2025) and guarantee fair evaluation, we report the canonical word-level **F1** and **Exact Match (EM)** scores. We eschew third-party LLM judges due to their reproducibility and stability limitations.

3.2 Baselines

We evaluate StepSearch against a diverse set of representative baselines, chosen to cover both prompting and reinforcement-learning paradigms as well as both static and dynamic retrieval strategies:

- **Naive Generation:** Direct generation and Chain-of-Thought (**CoT**) (Wei et al., 2022) reasoning;
- **RAG:** naive Retrieval-Augmented Generation (RAG) and **IRCoT** (Trivedi et al., 2023)

which use retrieval and the chain-of-thought (CoT) guide each other;

- **SFT:** Supervised fine-tuning (SFT) (Chung et al., 2024) without a search engine;

- **Large Reasoning Model:** RL-based fine-tuning (**R1**) (DeepSeek-AI et al., 2025) without a search engine and reasoning with in-process search (**Search-o1**) (Li et al., 2025);

- **Search with RL:** Existing outstanding reinforcement learning methods combined with external search engines including **Search-R1** (Jin et al., 2025), **ZeroSearch** (Sun et al., 2025) and **ReSearch** (Chen et al., 2025). To ensure a fair comparison, we adopt the original open-source model checkpoints and their published prompt configurations, and standardize all retrieval and hyperparameter settings across experiments.

3.3 Training Details

We conduct experiments on 4 models from Qwen family (Qwen et al., 2025): Qwen-2.5-7B (Base/Instruct) and Qwen-2.5-3B (Base/Instruct). In order to support the training of StepSearch, we generated our dataset with process supervision reference information based on the MuSiQue (Trivedi et al., 2022) dataset using the search key synthesis pipeline in Section 2.1.

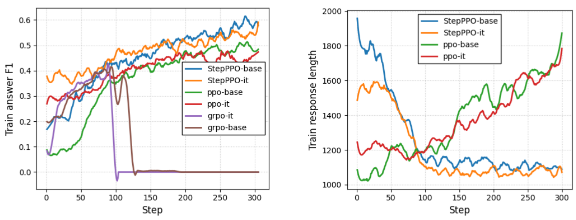
During training, we employ E5 (Wang et al., 2022) as the retriever over our synthesized dataset. For evaluation, we augment the corpus with the **2018 Wikipedia** dump (Karpukhin et al., 2020), as in Search-R1 (Jin et al., 2025), and uniformly retrieve $k = 3$ documents. Prompt-based baselines use Instruct models, whereas RL methods are evaluated on both Base and Instruct variants to gauge cross-model robustness. A more detailed experimental setup can be found in the Appendix B.

3.4 Main Results

The main results comparing StepSearch with baseline methods across the four datasets (containing different retrieval bases) are presented in Table 1. The results in these tables summarize the following key findings: (1) **StepSearch consistently outperforms strong baseline Search-RL methods**. Our method performance advantage holds for both in-domain multi-hop (*i.e.*, MuSiQue) and out-of-domain (*i.e.*, HotpotQA, 2WikiMultiHopQA, and Bamboogle) datasets, demonstrating the robustness of our method. (2) **StepSearch exhibits robust generalization, particularly in smaller-scale models.** Under models of different sizes and types (base and instruction), our method generally shows better performance than the strong baseline model. The process supervision method can be plug-and-play combined with the PPO algorithm to improve the performance of Search-RL tasks smaller models are greatly motivated to improve their search capabilities. (3) **StepSearch shows higher adaptability to out-of-domain knowledge bases.** Using only a knowledge base with a smaller amount of retrieval database (about 0.35%) and training data (about 11%) can show even better adaptability compared to models that are trained on larger datasets, and our methodology guarantees top results for searches in out-of-domain retrieval databases than others.

4 Further Analysis

4.1 Different RL Comparison



(a) Answer f1 score (b) Response length

Figure 4: StePPO vs. PPO.

Leveraging Qwen2.5-Base (3B/7B), we compare GRPO and PPO against StePPO. Results in Table 2 and training curves in Figure 4 show that **(1) StePPO drives high-quality**, low-cost generation (Figure 4b), achieving higher accuracy in fewer rounds with shorter outputs due to enriched search supervision; **(2) PPO-based algorithm delivers the most stable training**, while GRPO (both Base and Instruct) is prone to reward collapse at higher learning rates (Figure 4a); and **(3) StePPO attains the highest convergence speed and peak effectiveness**, outperforming both PPO and GRPO in final F1 (Figure 4a, Table 2).

4.2 Ablation Study

To further validate the effectiveness of StePPO's search process reward mechanism, we conducted extensive ablation experiments based on the

Method	HotpotQA†		2Wiki†		MuSiQue†		Bamboogle†		MuSiQue*	
	EM	*F1*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*
Qwen2.5-3b-Base/Instruct										
Naive Geneartion	0.145	0.237	0.249	0.356	0.018	0.079	0.030	0.086	-	-
RAG	0.251	0.359	0.221	0.316	0.051	0.135	0.076	0.161	-	-
SFT	0.191	0.299	0.248	0.356	0.039	0.110	0.112	0.181	-	-
IRCoT	0.171	0.241	0.196	0.265	0.059	0.123	0.245	0.356	-	-
R1-base	0.191	0.306	0.277	0.334	0.051	0.106	0.232	0.305	-	-
R1-instruct	0.210	0.299	0.288	0.389	0.065	0.136	0.213	0.317	-	-
Search-o1	0.240	0.326	0.207	0.309	0.045	0.117	0.316	0.436	-	-
ZeroSearch-base	0.260	0.354	0.234	0.281	0.056	0.116	0.096	0.193	0.072	0.151
ZeroSearch-instruct	0.265	0355	0.233	0.278	0.059	0.121	0.144	0.243	0.073	0.147
Search-R1-base†	0.272	0.361	0.248	0.296	0.081	0.146	0.176	0.270	0.118	0.193
Search-R1-instruct†	0.304	0.401	0.293	0.352	0.120	0.188	0.240	0.344	0.162	0.240
StepSearch-base*	0.329	0.434	**0.339**	**0.395**	**0.181**	**0.273**	0.328	0.419	**0.274**	**0.375**
StepSearch-instruct*	**0.345**	**0.452**	0.320	0.385	0.174	0.261	**0.344**	**0.452**	0.258	0.357
Qwen2.5-7b-Base/Instruct										
Naive Geneartion	0.187	0.291	0.246	0.352	0.027	0.083	0.123	0.242	-	-
SFT	0.196	0.175	0.269	0.374	0.054	0.131	0.110	0.203	-	-
IRCoT	0.141	0.232	0.142	0.241	0.072	0.159	0.216	0.319	-	-
RAG	0.287	0.391	0.231	0.226	0.061	0.142	0.214	0.316	-	-
R1-base	0.234	0.326	0.270	0.368	0.076	0.151	0.287	0.395	-	-
R1-instruct	0.241	0.345	0.287	0.392	0.079	0.154	0.284	0.397	-	-
Search-o1	0.193	0.288	0.181	0.289	0.053	0.127	0.302	0.427	-	-
ZeroSearch-base	0.294	0.394	0.275	0.324	0.102	0.175	0.258	0.373	0.134	0.218
ZeroSearch-instruct	0.325	0.432	0.309	0.370	0.120	0.204	0.267	0.409	0.184	0.280
Research-base*	0.294	0.388	0.264	0.313	0.143	0.230	0.373	0.449	0.206	0.309
Research-instruct*	0.362	0.471	0.354	0.416	0.184	0.271	0.424	0.544	0.250	0.348
Search-R1-base†	**0.432**	**0.547**	0.350	0.411	0.206	0.290	0.430	0.545	0.305	0.401
Search-R1-instruct†	0.394	0.502	0.312	0.376	0.181	0.262	0.384	0.501	0.268	0.352
StepSearch-base*	0.380	0.493	**0.385**	**0.450**	0.216	**0.324**	**0.467**	**0.573**	**0.346**	**0.461**
StepSearch-instruct*	0.386	0.502	0.366	0.431	**0.226**	0.312	0.400	0.534	0.339	0.443

Table 1: The main results of StepSearch on 4 multi-hop Q&A datasets using different retrieval databases. Search-R1 is trained based on NQ+HotpotQA dataset (170k) while ours and ReSearch are on MuSiQue (19k), "†" refers to train or test on wiki-18 knowledge base and "∗" for our customized base build on MuSiQue.

Method	HotpotQA		2Wiki		MuSiQue		Bamboogle	
	EM	*F1*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*
Qwen2.5-7b-Base								
StePPO	**0.380**	**0.493**	**0.385**	**0.450**	**0.216**	**0.324**	**0.467**	**0.573**
PPO	0.374	0.479	0.282	0.329	0.198	0.280	0.432	0.549
GRPO	0.351	0.462	0.266	0.345	0.202	0.291	0.400	0.512
Qwen2.5-3b-Base								
StePPO	**0.329**	**0.434**	**0.339**	**0.395**	**0.181**	**0.273**	**0.328**	**0.419**
PPO	0.223	0.315	0.225	0.273	0.090	0.163	0.176	0.266
GRPO	0.256	0.366	0.256	0.325	0.114	0.190	0.224	0.314

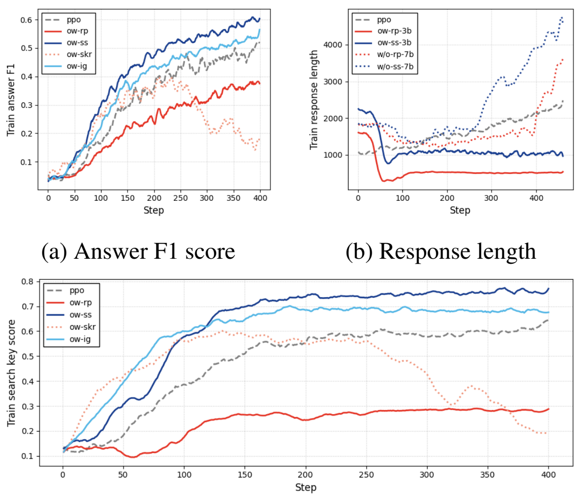
Table 2: Performance of models trained by different RL algorithms on multi-hop Q&A datasets. PPO and GRPO are trained on the reward of final answer F1. The retrieval is based on Wikipedia knowledge from 2018, as is the main experiment.

Qwen2.5-7B-Base model. The Table 3 below shows the evaluation of each configuration model on different datasets, and Figure 5. shows the experimental process record. The experimental results revealed these phenomena: **(1) StePPO has more prominent advantages in small parameter models** (Figure 5a, Table 3). Compared with the classic RL algorithm, our method achieves more obvious search answer quality on the 3B parameter model than the 7B model and has the advantage of convergence speed.; **(2) Redundancy penalty alone does not optimize the search ability** (*ow-rp* in Figure 5a), but it can force the model to perform high-quality, low-repetition effective search when applying information gain calculation (*ow-ss* in Figure 5a), thereby achieving a higher capacity ceiling; **(3) Searching keyword reward values can signif-**

icantly improve the model convergence speed (*ow-skr* in Figure 5a and 5c), but without process supervision, it may lead to hallucination and reward collapse problems more quickly, causing the model to respond incoherently and fail to converge; **(4) The effectiveness of fine-grained process rewards**, the step-wise token-level reward mechanism (*ow-ss*) has more obvious advantages and stability compared to the global reward of search process (*ow-skr*).

Method	HotpotQA *EM*	HotpotQA *F1*	2Wiki *EM*	2Wiki *F1*	MuSiQue *EM*	MuSiQue *F1*	Bamboogle *EM*	Bamboogle *F1*
Qwen2.5-7b-Base								
StePPO	0.380	0.493	0.385	0.450	**0.216**	**0.324**	**0.467**	**0.573**
w/o key reward	**0.404**	**0.528**	**0.388**	**0.468**	0.204	0.315	0.432	0.542
w/o penalty	0.377	0.494	0.300	0.367	0.190	0.286	0.392	0.502
w/o infon gain	0.365	0.468	0.3651	0.422	0.208	0.303	0.421	0.540
w/o steps reward	0.361	0.475	0.360	0.433	0.192	0.283	0.384	0.485
Qwen2.5-3b-Base								
PPO	0.228	0.315	0.225	0.273	0.090	0.163	0.176	0.266
ow key reward	0.259	0.375	0.178	0.282	0.127	0.218	0.232	0.334
ow penalty	0.258	0.364	0.227	0.279	0.083	0.177	0.192	0.312
ow info gain	0.323	0.432	**0.355**	**0.425**	0.169	0.249	0.344	0.439
ow steps reward	0.328	0.437	0.326	0.391	**0.185**	**0.282**	**0.360**	**0.487**
ow sub-answer	**0.339**	**0.448**	0.293	0.354	0.176	0.258	0.312	0.432

Table 3: Results on models 7b and 3b respectively. '*w/o*' represent 'with out' while '*ow*' for 'only with', 'sub-answer' represents a process supervision rewards based on intermediate sub-answers.



(a) Answer F1 score

(b) Response length

(c) Search key score

Figure 5: Training dynamics of correctness, response length, and search-key scores in ablation experiments. '*w/o*' represent 'with out' while '*ow*' for 'only with', '*rp*' stands for 'redundancy penalty','*ig*' for 'information gain', '*ss*' is 'step score' ('*ig*' + '*rp*') and '*skr*' means the global reward 'search key reward'

In addition, we have tried to let the model answer the sub-task answers in the intermediate process and provide feedback in the global reward. Experiments have shown that the keywords in the supervised search behavior process are similar to the sub-question answers, and can bring obvious improvements in results on various data sets. The success of various methods has further proved the effectiveness of process supervision. The prompt template can be found in the Appendix A.

4.3 Case Study

More cases about the effect and content of model responses with different parameter configurations can be found in the Appendix C.

5 Conclusion

We proposed StepSearch, a token-level and step-wise RL training method for search agents, with corresponding data pipeline to synthesize intermediate information. With supervision mechanisms of the search process of different granularities and a reward scheme combined with information gain theory, it ignites the ability of LLMs to handle multi-hop Q&A tasks by efficiently interacting with external search engines and achieved SOTA performance among search-RL methods. Extensive experiments have shown that StepSearch greatly improves search capabilities through the combination of reward and inhibition mechanisms, and its performance on multiple data sets exceeds that of existing search RL models by training on smaller datasets. In addition, the method in this article is applicable to both base and instruction-tuned models particularly effective for small ones.

6 Limitations

Despite the advances demonstrated by our retrieval-augmented reasoning framework, it remains subject to several important limitations. Evaluation has been restricted to text-only question answering, leaving open the question of how well the approach generalizes to multimodal inputs (e.g., images, audio) and to tasks that cross paradigmatic boundaries. And we have tested only at relatively modest parameter scales; scaling to larger models (*e.g.*, 14 B, 32 B) may exacerbate issues such as reward collapse and unstable training dynamics, necessitating novel stabilization and regularization strategies. Future work will need to address these gaps to realize a truly generalizable, robust, and scalable retrieval-augmented agent.

References

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. Preprint, arXiv:2503.19470.

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. Preprint, arXiv:2501.17161.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70):1–53.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. CoRR, abs/2501.12948.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. Preprint, arXiv:2404.16130.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. Preprint, arXiv:2207.06300.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. Preprint, arXiv:2410.12837.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2023. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. Advances in neural information processing systems, 36:45870–45894.

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

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Julia Hockenmaier, and Tong Zhang. 2025. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. Preprint, arXiv:2503.12759.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. Openai o1 system card. CoRR, abs/2412.16720.

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. Preprint, arXiv:2503.00223.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O. Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. Preprint, arXiv:2410.05983.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. Preprint, arXiv:2503.09516.

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), pages 6769–6781.

Joohyun Lee and Minji Roh. 2024. Multi-reranker: Maximizing performance of retrieval-augmented generation in the financerag challenge. Preprint, arXiv:2411.16732.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33:9459–9474.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. Preprint, arXiv:2501.05366.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Rithesh Murthy, Liangwei Yang, Zuxin Liu, Tian Lan, Ming Zhu, Juntao Tan, Shirley Kokane, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. 2024. Pract:

Optimizing principled reasoning and acting of llm agent. Preprint, arXiv:2410.18528.

Avinash Patil. 2025. Advancing reasoning in large language models: Promising methods and approaches. Preprint, arXiv:2502.03671.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. arXiv preprint arXiv:2210.03350.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2025. Tool learning with large language models: A survey. Frontiers of Computer Science, 19(8):198343.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. Preprint, arXiv:2412.15115.

Juan Ramos and 1 others. 2003. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, volume 242, pages 29–48. Citeseer.

Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. 2024. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), volume 24, page 155–161. IEEE.

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, 36:68539–68551.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. Preprint, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. Preprint, arXiv:2402.03300.

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

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. 2025. Tool learning in the wild: Empowering language models as automatic tool agents. Preprint, arXiv:2405.16533.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. Preprint, arXiv:2501.09136.

Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, Sharat Israni, Charlotte A Nelson, Sui Huang, and Sergio E Baranzini. 2024. Biomedical knowledge graph-optimized prompt generation for large language models. Preprint, arXiv:2311.17330.

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. Preprint, arXiv:2503.05592.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. 2025. Zerosearch: Incentivize the search capability of llms without searching. Preprint, arXiv:2505.04588.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. CoRR, abs/2501.12599.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. Transactions of the Association for Computational Linguistics, 10:539–554.

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), pages 10014–10037.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. Chain-of-retrieval augmented generation. CoRR, abs/2501.14342.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. Preprint, arXiv:2502.14768.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380.

Yufan Ye, Ting Zhang, Wenbin Jiang, and Hua Huang. 2025a. Process-supervised reinforcement learning for code generation. Preprint, arXiv:2502.01715.

Zihuiwen Ye, Luckeciano Carvalho Melo, Younesse Kaddar, Phil Blunsom, Sam Staton, and Yarin Gal. 2025b. Uncertainty-aware step-wise verification with generative reward models. Preprint, arXiv:2502.11250.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. Advances in Neural Information Processing Systems, 37:121156–121184.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. Preprint, arXiv:2503.12937.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. Preprint, arXiv:2402.19473.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. Preprint, arXiv:2504.03160.

Dawei Zhu, Xiyu Wei, Guangxiang Zhao, Wenhao Wu, Haosheng Zou, Junfeng Ran, Xun Wang, Lin Sun, Xiangzheng Zhang, and Sujian Li. 2025. Chain-of-thought matters: Improving long-context language models with reasoning path supervision. Preprint, arXiv:2502.20790.

Background You are a deep AI research assistant. I will give you a single-hop or multi-hop question. You don't have to answer the question now, but you should first think about your research plan or what to search for next. You can use search to fill in knowledge gaps.
Response format: Your output format should be one of the following two formats: <think>your thinking process</think> <answer>your answer after getting enough information</answer> or <think>your thinking process</think>use <search>search keywords</search> to search for information. For example, <think> plan to search: (Q1) (Q2) (Q3) ... /<think> <search> (Q1) question </search> <think> reasoning ... </think> <answer> Beijing </answer>..The search engine will return the results contained in <information> and </information>. Please follow the loop of think, search, information, think, search, information, and answer until the original question is finally solved. Note: The retrieval results may not contain the answer or contain noise. You need to tell whether there is a golden answer. If not, you need to correct the search query and search again. *Question:{question}*

Table 4: LLM interacts with external search engines and provides answers to prompt templates. The *{question}* in will be replaced with the actual question content.

A Prompt for Research Plan on Question Answering

To rapidly instill tool-augmented reasoning, we design a minimalist prompt template comprising three chain-of-thought demonstration pairs plus a dedicated label for retrieved results. Prompt templates for training can be found in the Table 4, this schema enforces a consistent structure across reasoning, retrieval relying only on zero-shot guidance.

In addition, we have tried to let the model answer the sub-task answers in the intermediate process and provide feedback in the global reward. Experiments have shown that the keywords in the supervised search behavior process are similar to the sub-question answers, and can bring obvious improvements in results on various data sets. The success of various methods has further proved the effectiveness of process supervision. The prompt template for this response can be found in the Table 5.

B Experiment Setups

Our implementation is based on Search-R1 (Jin et al., 2025), and our training is conducted using Verl (Sheng et al., 2024). Our experiments are carried out on two series of models: Qwen-2.5-3B and Qwen-2.5-7B (Qwen et al., 2025). The **MuSiQue** (Trivedi et al., 2022) training set processed through our pipeline is used for training, while the full **2WikiMultiHopQA** (Ho et al., 2020), **Bamboogle** (Press et al., 2022), **HotpotQA** (Yang et al., 2018), and **MuSiQue** test or validation sets are used for evaluation. EM and F1 score are employed as evaluation metrics.

We train for a total of 500 steps. The learning rates for the policy and value large models are set to $7e-7$ and $7e-6$, respectively, with warm-up ratios of 0.285 and 0.015 learning rate warm-up steps ratio. Training is conducted across two nodes with 16 H800 GPUs. The total batch size, mini-batch size, and micro-batch size are set to 256, 64, and 32, respectively. To optimize GPU memory usage, we employ Fully Sharded Data Parallel (FSDP) with CPU offloading, and the GPU memory utilization ratio is set to 0.7.

For rollout sampling, both the temperature and top_p are set to 1.0. The KL-divergence regularization coefficient β and the clipping ratio are set to $1e-3$ and 0.2, respectively.

C Case Study

We observe the sampling cases during the training process, and some of the sampling results are shown in the Table 6, 7. Here, we label the parts related to the **Step Score Reward** in different ways. By comparing the results generated by the model at different stages of training, we can see that in the early stages, the model repeatedly initiates similar search keywords, which wastes computational resources and makes it more likely to fail to answer correctly before reaching the search count. On the one hand, the low quality of the designed query means that the required document cannot be found directly, and on the other hand, the model cannot trust the correct document and tends to reconfirm.

In addition, we observed in our experiments that

Answer the given question. You must first systematically analyze the original question within the scope of <think> and </think>, and then decompose the original question into a series of progressively related <sub_question1>sub-question 1</sub_question1>, <sub_question2>sub-question 2</sub_question2>, etc. Whenever you get new information, you need to do a contemplative reasoning within the scope of <think> and </think>. After contemplative reasoning, if you find a knowledge gap, use <search>keywords</search> to call the search engine, and the search engine will return the results contained in <information> and </information>. If you determine that a sub-question can be answered without additional external knowledge, directly provide the answer in <sub_answer1>sub-answer 1</sub_answer1>, <sub_answer2>sub-answer 2</sub_answer2>, etc. without further explanation. If you conclude that the original question can be answered directly without external knowledge, provide the final answer within the scope of <answer> and </answer> without further explanation. Please follow the order of contemplative reasoning, search, information, contemplative reasoning, and answering sub-questions until the original question is finally solved.
Question:{question}

Table 5: LLM interacts with external search engines and provides answers to prompt templates. The *{question}* in will be replaced with the actual question content.

when the GLOBAL level of search key score was applied as the only process reward, the model was more likely to discover the trick of the format reward. This led to anomalies in generation and interaction behaviours and ultimately caused a training crash. As shown in the Table 9, the model issued a query request without using the correct search format and even made up information for phantom answers.

Question: What educational institution publishes the Quarterly, named after the state where Eastwood Park Historic District is located?

Ground Truth: ['University of North Dakota', 'UND']

StepSearch:

<think> My goal is to find the educational institution that publishes the Quarterly, which is named after the state where Eastwood Park Historic District is located. </think>

<search> Eastwood Park Historic District </search>

<information> Doc 1<## Title: Eastwood Park Historic District ##> The Eastwood Park Historic District is a historic district... Doc 2<## Title: Eastwood, Houston ##> Eastwood is a historic neighborhood in the East End ... Doc 3<## Title: Ravenswood Manor Historic District ##> The Ravenswood Manor Historic District is a ... </information>

<think> I found that Eastwood Park Historic District is located in Minot, North Dakota, but I need to identify the state's name to find the Quarterly's publisher. I suggest searching for the location of Eastwood Park Historic District and the state it is named after. </think>

<search> location of Eastwood Park Historic District </search>

<information> ~~Doc 1<## Title: Eastwood Park Historic District ##> The Eastwood Park Historic District is a historic district...~~ ~~Doc 2<## Title: Eastwood, Houston ##> Eastwood is a historic neighborhood ...~~ Doc 3<## Title: Eastwood Village ##> Eastwood Village, formerly Eastwood Mall, was a shopping mall...</information>

<think> The information now confirms that the Eastwood Park Historic District is in Minot, North Dakota. Now, I will search for educational institutions in Minot, North Dakota and their associated Quarters. </think>

<search> educational institutions in Minot, North Dakota </search>

<information> ~~Doc 1<## Title: Eastwood Park Historic District ##> The Eastwood Park Historic District is a historic district ...~~ Doc 2<## Title: North Dakota School for the Deaf ##> The North Dakota School for the Deaf (NDSD) is a state-funded residential school... Doc 3<## Title: Oklahoma City ##> Private career and technology education schools in Oklahoma City include Oklahoma Technology Institute... </information>

<think>The search results provide information about the North Dakota Quarterly, which is a quarterly literary journal published by the University of North Dakota. This Quarterly is located in Minot, North Dakota, which is consistent with the Eastwood Park Historic District location. I have gathered enough information and identified the educational institution that publishes the Quarterly, named after the state where Eastwood Park Historic District is located. </think>

<answer> University of North Dakota </answer>

Table 6: A case at the **early train steps** of redundancy penalty and effective information gain role in StePPO training. **Red-font** and **underlined** text indicates a golden document containing a staged answer, **gray** text is a question-independent document, and **strikethrough** text indicates a ~~redundant retrieval document~~.

Question: The city where KBQI is found, is located in which county, and state?

Ground Truth: ['Bernalillo County, New Mexico', 'Bernalillo County']

StepSearch:

<think> First, I need to identify the city where KBQI is located. Then, I will find the county and state in which that city is situated. </think>

<search> where is KBQI located </search>

<information> Doc 1<## Title: KBQI ##> KBQI (107.9 FM, "Big I 107.9") is a radio station in Albuquerque... Doc 2<## Title: KBIK ##> KBIK (102.9 FM) is a radio station licensed in Independence, Kansas... Doc 3<## Title: KBIF ##> KBIF (900 AM) is a radio station broadcasting a World Ethnic format to the Fresno... </information>

<think> From the information, I found out that KBQI is located in Albuquerque. Now I need to find the county and state in which Albuquerque is situated. </think>

<search> Albuquerque county and state </search>

<information> Doc 1<## Title: Bernalillo County Metropolitan Court ##> The Bernalillo County Metropolitan Court is the Judicial system of the metropolitan areas of Albuquerque, New Mexico and Bernalillo County, New Mexico. Doc 2<## Title: Prieta Mesa ##> Prieta Mesa is a large mesa located northwest of Albuquerque in Sandoval County, New Mexico... Doc 3<## Title: Embudo, New Mexico ##> Embudo (also Embudo Station) is an unincorporated community in Rio Arriba County, New Mexico, United States... </information>

<think> Albuquerque is located in Bernalillo County, New Mexico. </think>

<answer> Bernalillo County, New Mexico </answer>

Table 7: A case at the late training steps of redundancy penalty and effective information gain role in StePPO training. **Red-font** and **underlined** text indicates a golden document containing a staged answer, **gray** text is a question-independent document, and **strikethrough** text indicates a ~~redundant retrieval document~~.

Question: When was the leader who wanted to unify Germany (Prussia) born?

Ground Truth: ['1862']

StepSearch:

<think> First, I need to know who the leader was that wanted to unify Germany (Prussia). Then, I will search for the birth date of that leader. </think>

<search> Who is the leader that wanted to unify Germany (Prussia)? </search>

<information> Doc 1<## Title: Unification of Germany ##> Historians debate whether Otto von Bismarck... Doc 2<## Title: William I, German Emperor ##> William I, or in German Wilhelm I. (full name: William Frederick Louis... Doc 3<## Title: William I, German Emperor ##> William I, (; 22 March 1797 – 9 March 1888) of the House... </information>

<think> Based on the information, the leader who wanted to unify Germany (Prussia) was Otto von Bismarck. Now, I will search for the birth date of Otto von Bismarck. </think>

<search> When was Otto von Bismarck born? </search" **(!!! wrong format search action !!!)**

<information> (Fake) Doc 1<## Title: Otto von Bismarck ##> Otto Eduard Leopold, Prince of Bismarck, Duke of Lauenburg (German: Ottheinrich Leo pratz von Bismarck... </information> (Fake)

<think> I found that Otto von Bismarck was born on 1 April 1815. So, the answer is 1 April 1815. </think>

<search> When was Otto von Bismarck born? </search) **(!!! wrong format search action !!!)**

<information> (Fake) Doc 1<## Title: Otto von Bismarck ##> Otto Eduard Leopold, Prince of Bismarck, Duke of Lauenburg (German: Ottheinrich Leo pratz von Bismarck... </information> (Fake)

<search> When was Otto von Bismarck born? </search» **(!!! wrong format search action !!!)**

<search> When was Otto von Bismarck born? </search] **(!!! wrong format search action !!!)**

<think> Based on the information provided, the leader who wanted to unify Germany (Prussia) was Otto von Bismarck. Therefore, the answer is 1 April 1815. </think>

<answer> 1 April 1815 </answer>

Table 8: when only applied **Search Key Score** reward, models fail to correctly perform search interaction behaviors because of the trick of formatting rewards, and then make up their own data to cause illusory problems.

Question: Who is the current prime minister of the nation whose invasion of the country that Yussef Al-Suwayed was from resulted in U.S. military action?
Ground Truth: ['Haider al - Abadi']
<think>To answer the question of who the current prime minister of the nation that Yussef Al-Suwayed was from is, let's break down the information step by step. </think> <sub_question1>What country is Yussef Al-Suwayed from?</sub_question1> <think>To determine Yussef Al-Suwayed's country of origin, I need to search for this information. </think> <search>Who is Yussef Al-Suwayed and what country is he from?</search> <information> Doc 1<## Title: Yussef Al-Suwayed ##>Yussef Al-Suwayed (born 20 September 1958) is a Kuwaiti football midfielder ... Doc 2<## Title: Nutrition ##>The first recorded nutritional experiment with human subjects is found in the Bible's Book of Daniel... Doc 3<## Title: Samer Saeed ##>Samer Saeed Mujbel Al Mamoori (born December 1, 1987 in Iraq) is ... </information> <sub_answer1>Kuwait</sub_answer1>
<think> Now that I have determined Yussef Al-Suwayed is from Kuwait, I need to find out which nation invaded Kuwait and resulted in U.S. military action. </think> <sub_question2>Which nation invaded Kuwait and resulted in U.S. military action?</sub_question2> <think>To determine which nation invaded Kuwait and resulted in U.S. military action, I need to search for this information.</think> <search>Which nation invaded Kuwait and resulted in U.S. military action?</search> <information> Doc 1<## Title: Invasion of Kuwait ##>The Invasion of Kuwait on 2 August 1990 was a two - day operation conducted by Iraq against the neighboring state of Kuwait... Doc 2<## Title: Economic history of Germany ##>From the 1830s and 1840s, Prussia, Saxony, and other states ... Doc 3<## Title: Gulf War ##>The Gulf War (2 August 199028 February 1991), codenamed Operation Desert Shield ... </information> <sub_answer2>Iraq</sub_answer2>
<think>Now that I have determined that Iraq invaded Kuwait and resulted in U.S. military action, I can answer the original question. </think> <sub_question3>Who is the current prime minister of Iraq?</sub_question3> <think>To determine the current prime minister of Iraq, I need to search for this information.</think> <search>Who is the current prime minister of Iraq?</search> <information> Doc 1<## Title: Prime Minister of Iraq ##>The Prime Minister of Iraq is Iraq's head of government... Doc 2<## Title: 2018 Malaysian general election ##>This marked a historic defeat for the ruling Barisan Nasional coalition, which had been the governing party of Malaysia and its predec essor state, Malaya, since the country's independence in 1957... Doc 3<## Title: Prime Minister of Israel ##>The Prime Minister of Israel... </information> <think>Now I can answer the original question.</think> <answer>Haider al - Abadi<answer>

Table 9: Reward only **Sub_Answer** in process supervision