
An Empirical Study on Reinforcement Learning for Reasoning-Search Interleaved LLM Agents

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Reinforcement learning (RL) has demonstrated strong potential in training large
2 language models (LLMs) capable of complex reasoning for real-world problem
3 solving. More recently, RL has been leveraged to create sophisticated LLM-based
4 search agents that adeptly combine reasoning with search engine use. While the
5 use of RL for training search agents is promising, the optimal design of such
6 agents remains not fully understood. In particular, key factors—such as (1) reward
7 formulation, (2) the choice and characteristics of the underlying LLM, and (3) the
8 role of the search engine in the RL process—require further investigation. In this
9 work, we conduct comprehensive empirical studies to systematically investigate
10 these and offer actionable insights. We highlight several key findings: format
11 rewards are effective in improving final performance, whereas intermediate retrieval
12 rewards have limited impact; the scale and initialization of the LLM (general-
13 purpose vs. reasoning-specialized) significantly influence RL outcomes; and the
14 choice of search engine plays a critical role in shaping RL training dynamics and
15 the robustness of the trained agent during inference. These establish important
16 guidelines for successfully building and deploying LLM-based search agents in
17 real-world applications.

18 1 Introduction

19 Large language models (LLMs) [57] have demonstrated exceptional performance across a range
20 of natural language processing tasks, including question answering [26], summarization [56], and
21 open-ended text generation [8]. Recently, inspired by developments such as DeepSeek-R1 [9], rein-
22 forcement learning (RL) [14, 39] has been increasingly applied to LLMs to unlock more advanced
23 reasoning capabilities [45]. LLMs trained via RL have shown strong performance in tasks requiring
24 logical reasoning [49] and visual understanding [55], with promising applications emerging in spe-
25 cialized domains such as finance [25] and medicine [19]. However, these models often remain limited
26 to self-contained reasoning and lack the ability to interact with external environments or leverage
27 external tools. To address this, recent work has explored using RL to train LLMs as interactive agents,
28 capable of engaging with external environments and invoking tools, as demonstrated in multi-turn
29 game tasks [44] and user interface control tasks [24, 48].

30 A key agentic application for LLMs is *search*, where models decompose complex problems, perform
31 multi-turn reasoning, and iteratively interact with search engines to retrieve relevant information.
32 Prior work has explored prompt-based approaches [11, 41] and supervised fine-tuning (SFT) methods
33 [3, 34] to equip LLMs with search capabilities. However, these approaches face key limitations:
34 LLMs typically lack strong search proficiency from pretraining alone, and SFT requires costly manual
35 annotation of intermediate reasoning trajectories, making it challenging to scale. In contrast, recent
36 studies [4, 13, 37, 58] demonstrate that *RL with outcome-based rewards* offers an effective alternative

for training LLMs to perform reasoning and search in an interleaved manner—forming what is referred to as an *LLM-based search agent*. This training paradigm enhances the model’s ability to interact with search engines while eliminating the need for explicit supervision of intermediate reasoning steps, thus enabling scalable and more flexible agent learning.

While recent RL-based methods have demonstrated the potential to train effective LLM-based search agents, several key questions remain underexplored: (1) *How does reward design affect search agent training?* Although prior work [13] shows that outcome-based rewards alone can activate reasoning and search capabilities, it is unclear whether auxiliary rewards such as format rewards (which signal adherence to the agentic action format [9]) or intermediate retrieval rewards (which iteratively incentivize outcome-relevant retrievals [22]) can further enhance performance. (2) *How does the backbone LLM influence RL dynamics?* As suggested by [6], the choice of the base model is critical. Factors such as model scale (e.g., 3B vs. 32B) and type (e.g., general-purpose vs. reasoning-specialized) can significantly impact the learning dynamics. (3) *How does the search engine choice affect the learned agent?* This includes understanding how the quality of different search engines influence RL training dynamics and whether the resulting agent remains robust when the retrieval system is changed at inference time.

In this paper, we conduct comprehensive empirical studies to address the aforementioned research questions. Our key findings are summarized as follows: (1) *Reward Design*. We observe that incorporating a format reward significantly improves performance, particularly when training from a base LLM rather than an instruction-tuned one. In contrast, intermediate retrieval rewards do not yield consistent performance improvements, suggesting limited utility. (2) *Underlying LLM Backbone*. General-purpose LLMs outperform reasoning-specialized LLMs in RL settings, likely due to the latter’s weaker instruction-following capabilities at the early stages of training. Furthermore, scaling up the backbone model generally enhances final performance, although with diminishing returns. (3) *Search Engine Choice*. The quality of the search engine used during training strongly influences RL dynamics. Training with a non-informative search engine (e.g., random noise) leads the agent to avoid retrieval altogether, while a weak engine (e.g., BM25 [33]) results in frequent but less efficient search calls. In contrast, strong engines (e.g., dense retrievers) yield more stable learning. At inference time, the search agent is generally robust to diverse retrieval systems, and stronger search engines consistently lead to better downstream performance.

2 Related Works

2.1 Large Language Models and Reinforcement Learning

RL [14, 39] offers a principled framework for sequential decision-making, where an agent optimizes its behavior by interacting with an environment and maximizing cumulative rewards. In the context of LLM tuning, RL was popularized by Reinforcement Learning from Human Feedback (RLHF) [12, 17, 28], which first trains a reward model from human preference data [20] and then fine-tunes the policy LLM via Proximal Policy Optimization (PPO). While PPO enables high-quality alignment, it incurs significant computational overhead due to iterative optimization steps. Recent efforts to strike a better balance include Group Relative Policy Optimization (GRPO) [35], which removes the dependency on a learned value function by leveraging group-based baseline estimation, and RLOO [2], a simplified variant of REINFORCE [47] tailored for LLM training. More recently, DAPO [53] extends GRPO by introducing four key innovations tailored for large-scale LLM reinforcement learning: clip-higher reward capping, dynamic sampling for adaptive data efficiency, a token-level objective for finer-grained supervision, and overlong reward shaping to handle extended sequences. In parallel, VAPO [54] builds upon PPO by proposing a value-model-augmented framework, incorporating value pretraining, a decoupled Generalized Advantage Estimator (GAE), and an auxiliary language modeling loss on positive examples to improve credit assignment and stability. While these advancements have significantly enhanced the scalability and efficiency of RL-based LLM tuning, their application to LLM-driven search and reasoning tasks remains underexplored, highlighting a critical direction for future exploration.

2.2 Large Language Models as Search Agents

LLMs [1, 40, 57] have demonstrated strong reasoning capabilities [9] but often struggle with hallucinations and insufficient domain-specific knowledge [21, 29]. To address these, recent efforts explore

integrating LLMs with search engines to enable dynamic access to external knowledge. A prominent direction is to treat search engines as interactive tools that LLMs can call during inference [34]. This search-as-a-tool paradigm allows models to iteratively formulate queries, retrieve relevant content, and revise their responses based on external evidence [41]. Prompt-based methods such as IRCOT [41] and ReAct [52] enable interleaved reasoning and retrieval, while Toolformer [34] and self-RAG [3] uses supervised fine-tuning to learn when and how to call a search engine. However, these methods often depend on high-quality demonstration data, which is difficult to scale. As an alternative, RL offers a scalable and data efficient solution. Inspired by [9], recent works [4, 13, 37, 58] show that LLMs can acquire complex reasoning and search behaviors through RL only using outcome-based rewards. Despite this promise, there is still a lack of in-depth empirical study of different design choices in RL for LLM search agents.

3 Preliminary

Reasoning-Search Interleaved LLM Agent (i.e., LLM-based Search Agent) [13, 36]. We consider an agentic LLM that performs interleaved, *multi-turn* reasoning and search engine interactions. In each iteration, the LLM-based search agent first engages in *reasoning* to analyze the current context and identify what additional information is needed. It then formulates a search query to *retrieve* relevant external information, which is incorporated into the *context* for subsequent reasoning. This iterative process continues until the model determines that sufficient information has been gathered to produce a final *answer*. The overall interaction follows a multi-turn reasoning–search loop: reasoning $\rightarrow$ search $\rightarrow$ context $\rightarrow$ reasoning $\rightarrow$ search $\rightarrow \dots \rightarrow$ reasoning $\rightarrow$ answering. To facilitate this process [52], the reasoning steps are enclosed within `<think> </think>`, search queries are wrapped in `<search> </search>`, retrieved information is inserted into `<information> </information>`, and the final answer is placed within `<answer> </answer>`.

RL for Training an LLM-based Search Agent. In [13, 58], they propose an RL objective to explicitly incorporate a search engine $\mathcal{R}$ during optimization for LLM search agent training. The objective is formalized as:

$$\max_{\pi_{\theta}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(\cdot | x; \mathcal{R})} [r_{\phi}(x, y)] - \beta \mathbb{D}_{\text{KL}} [\pi_{\theta}(y | x; \mathcal{R}) \| \pi_{\text{ref}}(y | x; \mathcal{R})], \quad (1)$$

where π_{θ} denotes the trainable policy, π_{ref} is a fixed reference model, r_{ϕ} represents the reward function, and $\mathbb{D}_{\text{KL}}$ denotes the KL divergence. Here, x are sampled from the dataset $\mathcal{D}$, and y denote the output sequence interleaving reasoning steps with search engine retrievals.

In contrast to prior approaches that generate rollouts exclusively from the model $\pi_{\theta}(\cdot | x)$ [28, 31], [13, 58] augment the generation process by interleaving retrievals via $\pi_{\theta}(\cdot | x; \mathcal{R})$, which can be interpreted as $\pi_{\theta}(\cdot | x) \otimes \mathcal{R}$, where $\otimes$ denotes a retrieval-reasoning composition.

The reward function r_{ϕ} serves as the primary optimization signal. [13, 58] employ a rule-based reward system focusing exclusively on **final outcome rewards**, which evaluate the correctness of the final answer. In factual reasoning tasks, correctness is assessed using exact string match (EM) evaluation: $r_{\phi}(x, y) = \text{EM}(a_{\text{pred}}, a_{\text{gold}})$, where a_{pred} is the predicted final answer extracted from the model’s response y , and a_{gold} is the ground-truth answer. In other words,

$$r_{\phi}(x, y) = \begin{cases} 1 & \text{if } a_{\text{pred}} = a_{\text{gold}}, \\ 0 & \text{if } a_{\text{pred}} \neq a_{\text{gold}}, \end{cases} \quad (2)$$

Although prior methods have demonstrated strong performance, there remains a notable gap in empirical studies systematically evaluating key design choices—specifically, the effectiveness of different reward formulations, the influence of underlying LLM characteristics, and the impact of search engine selection—on the reinforcement learning process for training search agents.

4 RL Rewards for LLM-based Search Agents

In [13, 58], an outcome-driven reward (i.e., string exact match) is employed through the RL process to guide the LLM on learning the reasoning and interleaved search engine calling functionality. However, in search scenarios, the LLMs need to follow a specific format in order to call the search engine (i.e., format reward) and the relevance of the intermediate search results can also guide the LLM on generating the proper queries (i.e., intermediate retrieval reward).

Table 1: Empirical study of the format reward. *Outcome only* refers to the RL variant with only the outcome reward. Base/Instruct refer to the version of the underlying LLM. $\lambda_f = 0.2$ for 3B/14B and $\lambda_f = 0.4$ for 7B. The best performance is set in bold. † / * represents in-domain/out-domain datasets.

Methods		General QA			Multi-Hop QA				Avg.
		NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2wiki [*]	Musique [*]	Bamboogle [*]	
Qwen2.5-3B-Base/Instruct									
PPO	Outcome only (base)	0.406	0.587	0.435	0.284	0.273	0.049	0.088	0.303
	w. format reward	0.428	0.607	0.459	0.371	0.387	0.150	0.323	0.389
	Outcome only (instruct)	0.341	0.545	0.378	0.324	0.319	0.103	0.264	0.325
	w. format reward	0.356	0.557	0.393	0.327	0.314	0.122	0.266	0.334
GRPO	Outcome only (base)	0.421	0.583	0.413	0.297	0.274	0.066	0.128	0.312
	w. format reward	0.429	0.602	0.435	0.372	0.383	0.148	0.307	0.382
	Outcome only (instruct)	0.397	0.565	0.391	0.331	0.310	0.124	0.232	0.336
	w. format reward	0.346	0.552	0.371	0.297	0.300	0.098	0.266	0.319
Qwen2.5-7B-Base/Instruct									
PPO	Outcome only (base)	0.480	0.638	0.457	0.433	0.382	0.196	0.432	0.431
	w. format reward	0.488	0.644	0.469	0.436	0.412	0.187	0.403	0.434
	Outcome only (instruct)	0.393	0.610	0.397	0.370	0.414	0.146	0.368	0.385
	w. format reward	0.383	0.593	0.399	0.376	0.317	0.151	0.371	0.370
GRPO	Outcome only (base)	0.395	0.560	0.388	0.326	0.297	0.125	0.360	0.350
	w. format reward	0.458	0.632	0.442	0.412	0.404	0.180	0.411	0.420
	Outcome only (instruct)	0.429	0.623	0.427	0.386	0.346	0.162	0.400	0.396
	w. format reward	0.393	0.609	0.397	0.367	0.344	0.147	0.387	0.378
Qwen2.5-14B-Base/Instruct									
PPO	Outcome only (base)	0.486	0.676	0.480	0.468	0.470	0.241	0.528	0.479
	w. format reward	0.499	0.680	0.472	0.452	0.431	0.215	0.468	0.459
	Outcome only (instruct)	0.424	0.660	0.442	0.436	0.379	0.210	0.480	0.433
	w. format reward	0.449	0.682	0.466	0.447	0.422	0.224	0.500	0.456
GRPO	Outcome only (base)	0.415	0.680	0.488	0.451	0.461	0.230	0.508	0.462
	w. format reward	0.500	0.693	0.500	0.481	0.488	0.261	0.516	0.491
	Outcome only (instruct)	0.482	0.667	0.434	0.429	0.424	0.191	0.492	0.446
	w. format reward	0.488	0.677	0.482	0.455	0.470	0.211	0.516	0.471

4.1 Format Reward

Motivation. When training an LLM-based search agent capable of reasoning and invoking external search engines, it is common to adopt the reasoning-action-observation workflow [52], where relevant content is wrapped within special tokens such as `<think>` `</think>`, `<search>` `</search>`, and `<information>` `</information>`. For instance, if the LLM fails to correctly format its search queries using `<search>` `</search>`, it cannot successfully trigger the search engine and retrieve the external information needed for problem solving. Thus, adhering to the prescribed format is critical for ensuring the effectiveness of the search agent. In this section, we explore how incorporating a format reward influences the RL training process of a search agent.

Experimental Design. In addition to the outcome reward defined in [13, 58], we introduce a format reward, resulting in the final reward function $r_\phi(x, y)$:

$$r_\phi(x, y) = \begin{cases} 1 & \text{if } a_{\text{pred}} = a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True}, \\ 1 - \lambda_f & \text{if } a_{\text{pred}} = a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{False}, \\ \lambda_f & \text{if } a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True}, \\ 0 & \text{if } a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{False}, \end{cases} \quad (3)$$

where $f_{\text{format}}(\cdot)$ verifies whether the response y follows the correct reasoning-action-observation format, including the appropriate use of special tokens. We assign a reward of λ_f when the LLM generates an incorrect answer in the correct format, and a reward of $1 - \lambda_f$ when the answer is correct but the format is incorrect. Details of the $f_{\text{format}}(\cdot)$ implementation are provided in Appendix C. We follow [13] for the training and testing datasets and use exact match as the outcome reward. Detailed experimental settings can be found in Appendix G.1.

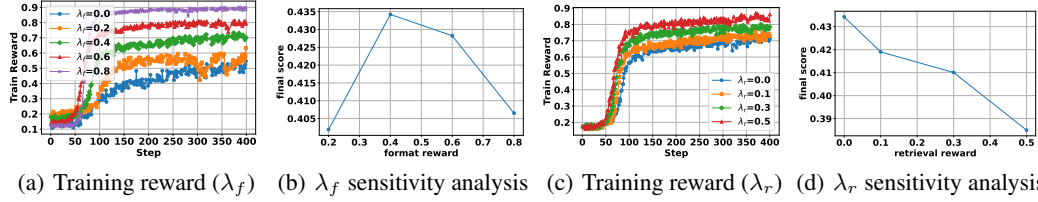


Figure 1: Empirical analyses on format reward and intermediate retrieval reward. (a) **Training reward curves** with varying **format reward** scaling factors (λ_f); larger λ_f values lead to faster convergence. (b) **Impact of λ_f** on final model performance; a small λ_f is ineffective, while an excessively large λ_f may cause overfitting to format reward. (c) **Training reward curves** under different **intermediate retrieval reward** scaling factors (λ_r); varying λ_r has limited effect on learning dynamics. (d) **Effect of λ_r** on final model performance; increasing λ_r degrades performance, suggesting that intermediate retrieval rewards are unnecessary, as the outcome reward sufficiently encourages effective query formulation. (LLM: Qwen2.5-7B-Base; RL Algorithm: PPO)

Results. Table 1 reports results across various datasets, LLM sizes, and RL algorithms. Detailed studies on λ_f using Qwen2.5-7B-Base and PPO are presented in Figures 1(a) and (b). We summarize the key findings as follows: (1) Adding a format reward consistently improves final model performance, particularly for base LLMs. This is because base LLMs lack strong instruction-following capabilities for search engine invocation, and the format reward helps mitigate this limitation. (2) Format reward accelerates RL convergence; larger λ_f values lead to faster convergence by explicitly guiding the model to issue correctly formatted search queries and interpret results effectively. (3) The choice of λ_f significantly impacts final performance. While a small λ_f is ineffective, an excessively large λ_f may cause overfitting, ultimately degrading final performance.

4.2 Intermediate Retrieval Reward

Motivation. Beyond the outcome reward, which directly evaluates the correctness of the final answer after multiple search interactions, it is possible to incorporate intermediate retrieval rewards that assess the quality of the retrieved documents during each search step [22]. By assigning positive rewards to cases where relevant information is retrieved, the LLM can be encouraged to generate higher-quality queries that yield more relevant retrieval results [22]. We investigate whether introducing intermediate retrieval rewards benefits the RL training process of LLM-based search agents.

Experimental Design. Building upon the outcome reward from [13, 58] and the format reward introduced in Section 4.1, we incorporate a retrieval correctness component, resulting in the following final reward function $r_\phi(x, y)$:

$$r_\phi(x, y) = \begin{cases} 1 & \text{if } a_{\text{pred}} = a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True}, \\ 1 - \lambda_f & \text{if } a_{\text{pred}} = a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{False}, \\ \lambda_f + \lambda_r & \text{if } a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True} \wedge f_{\text{ret}}(y) = \text{True}, \\ \lambda_f & \text{if } a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True} \wedge f_{\text{ret}}(y) = \text{False}, \\ 0 & \text{if } a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{False}, \end{cases} \quad (4)$$

where $f_{\text{ret}}(\cdot)$ determines whether the retrieved documents are relevant to the ground truth answer. The retrieved information can only be extracted when the rollout sequence follows the desired format, as described in Section 4.1 and the purpose of the intermediate retrieval reward is to provide a positive learning signal even when the final answer is incorrect. Thus, we introduce an additional reward term λ_r when $a_{\text{pred}} \neq a_{\text{gold}} \wedge f_{\text{format}}(y) = \text{True}$. In our experiments, we focus on short-form QA datasets, so we apply substring exact match as $f_{\text{ret}}(\cdot)$, following [11, 23], to evaluate whether ground truth appears in retrieved passages. Under this setting, even if LLM fails to generate correct final answer, it can still receive a positive reward for issuing effective queries that retrieve relevant documents. Detailed experimental settings can be found in Appendix G.2.

Results. Performance comparisons with and without intermediate retrieval rewards are presented in Table 2. The effect of varying λ_r is illustrated in Figures 1(c) and (d). Key observations include: (1) Adding intermediate retrieval rewards does not significantly improve final performance for either PPO

Table 2: Study of the intermediate retrieval reward. $\lambda_r = 0.1$. The best performance is set in bold. † / * represents in-domain/out-domain datasets.

Methods		General QA			Multi-Hop QA				Avg.
		NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2wiki [*]	Musique [*]	Bamboogle [*]	
Qwen2.5-3B-Base									
PPO	w.o. retrieval reward	0.428	0.607	0.459	0.371	0.387	0.150	0.323	0.389
	w. retrieval reward	0.405	0.567	0.407	0.326	0.330	0.104	0.242	0.340
GRPO	w.o. retrieval reward	0.429	0.602	0.435	0.372	0.383	0.148	0.307	0.382
	w. retrieval reward	0.434	0.605	0.433	0.379	0.378	0.142	0.323	0.385
Qwen2.5-7B-Base									
PPO	w.o. retrieval reward	0.488	0.644	0.469	0.436	0.412	0.187	0.403	0.434
	w. retrieval reward	0.472	0.629	0.452	0.436	0.402	0.180	0.363	0.419
GRPO	w.o. retrieval reward	0.458	0.632	0.442	0.412	0.404	0.180	0.411	0.420
	w. retrieval reward	0.453	0.628	0.450	0.416	0.375	0.164	0.387	0.410

Table 3: Performance of general LLM and reasoning LLM trained with RL on search agent task. The best performance is set in bold. † / * represents in-domain/out-domain datasets.

Methods	General QA			Multi-Hop QA				Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2wiki [*]	Musique [*]	Bamboogle [*]	
DeepSeek-R1-Distill-Qwen-7B								
PPO	0.389	0.542	0.402	0.334	0.326	0.122	0.290	0.344
GRPO	0.061	0.155	0.068	0.098	0.194	0.010	0.113	0.100
Qwen2.5-7B-Base								
PPO	0.488	0.644	0.469	0.436	0.412	0.187	0.403	0.434
GRPO	0.458	0.632	0.442	0.412	0.404	0.180	0.411	0.420

or GRPO. This may be attributed to the outcome reward already providing sufficient learning signal for generating effective queries, as a successful search engine call that retrieves relevant information directly contributes to producing the correct answer and receiving a positive reward. In contrast, the substring EM-based intermediate retrieval reward may overly constrains the retrieval trajectory and thus deviates the naturally learned trajectory from the outcome reward. (2) Varying λ_r has limited impact on learning dynamics. Increasing λ_r consistently leads to degraded performance, suggesting that intermediate retrieval rewards are unnecessary. The outcome reward alone is sufficient to encourage effective query formulation and downstream task success.

5 The Impact of Underlying Backbone LLM

In this section, we study how the choice of the LLM influences RL training for LLM-based search agents. Our investigation centers on two key characteristics of the base LLM: (1) type (*i.e.*, general-purpose vs. reasoning-optimized), and (2) scale (*i.e.*, 3B, 7B, 14B, and 32B).

5.1 Study of LLM types

Motivation. Effective training of LLM-based search agents via RL requires the LLM to possess two fundamental capabilities: *instruction following* and *reasoning*. Instruction following enables the model to learn how to properly issue search engine calls in the correct format, while reasoning equips the model to analyze retrieved information and solve complex problems. However, it remains underexplored whether general-purpose or reasoning-specialized LLMs provide a more suitable foundation for RL-based training.

Experimental Design. We follow the experimental setup in [13] and conduct RL training on two LLM variants: (1) Qwen2.5-7B-Base [50], a general-purpose 7B parameter pretrained LLM, and (2) DeepSeek-R1-Distill-Qwen-7B [9], a 7B reasoning-specialized model distilled from DeepSeek-R1. Both models are trained under identical conditions to ensure a fair comparison. Detailed experimental settings and results on 14B LLMs can be found in Appendix G.3 and F.

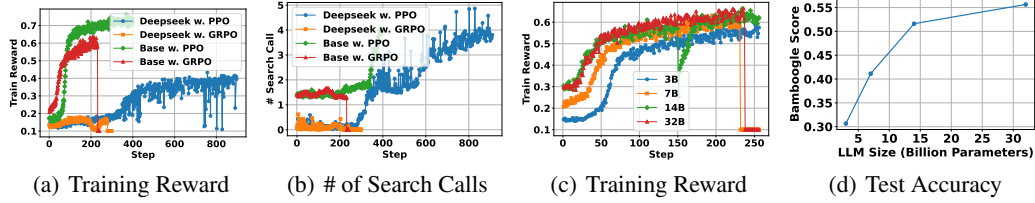


Figure 2: The study of underlying pretrained LLM for development of LLM-based search agents with RL. (a) **Training reward with different type of LLMs** - general-purpose LLM (Qwen2.5-7B-Base) and reasoning LLM (DeepSeek-R1-Distill-Qwen-7B). We observe that general-purpose LLM performs better than reasoning LLMs with both PPO and GRPO. (b) **# of Search engine calls with different type of LLMs**: General LLM learns to call the search engine faster than the reasoning LLM. This potentially stems from the fact the general LLMs are better for following instructions. (c) **Training reward with different size of LLMs**: Larger LLMs can lead to higher training reward. (d) **Test accuracy with different size of LLMs**: On the challenging Bamboogle dataset [30], the performance increases consistently as the LLM size increases.

209 **Results.** The training reward and search engine call frequency curves are presented in Figures 2(a)
 210 and 2(b), respectively. Final performance results are summarized in Table 3. We observe the following
 211 key findings: (1) The RL training process is more stable and effective when initialized with the
 212 general-purpose LLMs compared to the reasoning-specialized ones. This suggests that the general-
 213 purpose ones already possess sufficient basic reasoning capabilities to support the search agent task
 214 without requiring specialized pretraining. (2) The reasoning LLM struggles to initiate search engine
 215 calls during the early stages of training, leading to insufficient exploration. In the absence of positive
 216 reward signals from successful rollouts involving search, the model fails to consistently learn to
 217 engage with the search engine. This behavior primarily stems from the reasoning LLMs’ limited
 218 instruction-following capabilities, which hinder their ability to learn the correct format for invoking
 219 the search API. (3) While the reasoning LLMs eventually learn to perform interleaved reasoning and
 220 retrieval when trained with PPO, this progress is slow and gradual. In contrast, training with GRPO
 221 leads to training collapse. We attribute this to PPO’s lower variance and more stable policy updates,
 222 which better support the complex exploration required for search-augmented reasoning tasks.

223 5.2 The Scale Up of LLM-based Search Agent

224 **Motivation.** Prior work has demonstrated that LLM capabilities improve predictably with increased
 225 model size, as described by scaling laws [10, 15]. However, it remains unclear whether similar scaling
 226 behavior holds when LLMs are further RL-tuned as search agents. Specifically, does increasing
 227 model size consistently improve the agent’s ability to reason and interact with search engines?

228 **Experimental Design.** We evaluate scaling laws using RL with both outcome-based rewards and
 229 additional format rewards, as introduced in Section 4.1. Following the experimental setup in [13],
 230 we train Qwen2.5 models of varying sizes (3B, 7B, 14B, 32B) on the NQ and HotpotQA training
 231 datasets using the GRPO algorithm. We use a fixed learning rate of 5×10^{-7} , and evaluate on the
 232 out-of-distribution Bamboogle dataset. Detailed experimental settings can be found in Appendix G.4.

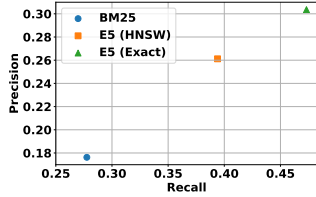
233 **Results.** As shown in Figure 2(c), the training reward consistently improves with increasing LLM
 234 size, indicating that larger models are better able to learn effective reasoning and search engine usage.
 235 The corresponding inference performance is presented in Figure 2(d). While test performance also
 236 improves with model size, the rate of improvement diminishes. This suggests that the search agent
 237 task, unlike pure language modeling, relies less on parametric knowledge stored in large LLMs, and
 238 more on effective external information acquisition through retrieval.

239 6 Improved LLM-based Search Agents with Stronger Search Engines

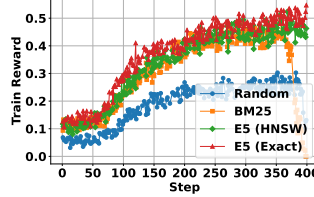
240 The choice of search engine plays a critical role in determining retrieval quality, which in turn
 241 influences both the RL training dynamics and the inference-time performance of the LLM-based
 242 search agent. During training, higher-quality search engines that provide more relevant information
 243 can encourage the agent to achieve its objectives with fewer search calls, as the retrieved content more

Table 4: Final performance with different search engine for both training and inference. The best performance is set in bold. (LLM: Qwen2.5-7B-Base; RL: PPO)

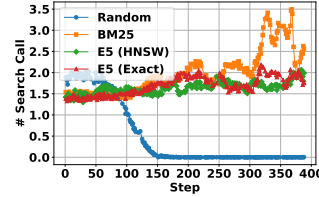
Engine	NQ		TriviaQA		PopQA		HotpotQA		2wiki		Musique		Bamboogle		Avg.	
	Recall	EM	Recall	EM	Recall	EM	Recall	EM	Recall	EM	Recall	EM	Recall	EM	Recall	EM
Random	0.000	0.237	0.000	0.494	0.000	0.177	0.000	0.217	0.000	0.269	0.000	0.058	0.000	0.234	0.000	0.241
BM25	0.216	0.341	0.445	0.607	0.255	0.322	0.273	0.404	0.216	0.370	0.076	0.137	0.061	0.280	0.176	0.352
E5 (HNSW)	0.436	0.468	0.509	0.621	0.304	0.366	0.237	0.372	0.146	0.287	0.092	0.137	0.104	0.400	0.261	0.379
E5 (Exact)	0.462	0.481	0.561	0.638	0.423	0.457	0.276	0.433	0.198	0.382	0.098	0.196	0.107	0.424	0.304	0.430



(a) Search engine performance



(b) Training reward



(c) # Search calls

Figure 3: Effect of Search Engine Choice on RL Training Dynamics. (a) **Retrieval Quality Ranking:** E5 (Exact) > E5 (HNSW) > BM25 > Random. (b) **Training Stability and Performance:** Stronger search engines (e.g., E5 Exact, E5 + HNSW) lead to more stable training and higher final performance, while weaker engines (e.g., Random, BM25) achieve suboptimal outcomes. (c) **Search Engine Usage Behavior:** With Random Noise, the agent quickly learns to avoid using the search engine. With BM25, the agent gradually increases search calls to compensate for limited retrieval quality. With E5, the agent issues search calls more strategically, reflecting more efficient search behavior.

effectively supports reasoning and decision-making. In contrast, lower-quality search engines that return less relevant information may lead the agent to either over-rely on its internal knowledge or issue multiple search queries to compensate for inadequate results. At inference time, the quality of the retrieved information directly impacts the agent’s ability to generate accurate and useful responses. In the following sections, we systematically investigate the effects of search engine choice on both the training and inference stages of search-augmented LLMs.

6.1 Training with Different Search Engines.

Motivation. During RL training, the LLM-based agent learns to interact with the search engine and receive positive reward feedback while it solves problems using the retrieved relevant information. A *strong* search engine provides more relevant results, leading to consistent positive outcome rewards. Consequently, the LLM learns to solve problems with fewer search calls. In contrast, a *weak* search engine discourages reliance on retrieval or forces the agent to issue multiple search queries to compensate for low-quality results. We empirically investigate how different search engines influence the RL training dynamics of an LLM-based search agent.

Experimental Design. We conduct experiments using the Qwen2.5-7B-Base model as the LLM and Proximal Policy Optimization (PPO) as the RL algorithm. Four search engine configurations are explored with the Wikipedia-18 corpus [16]: (1) **Random Noise:** Returns randomly selected passages for a given query. (2) **BM25** [33]: A sparse retrieval method based on exact token matching and term frequency. (3) **E5 (HNSW)** [27, 42]: A dense retrieval method that encodes queries and passages into semantic embeddings, using dot product similarity for matching. HNSW provides efficient approximate nearest neighbor (ANN) search at the cost of some accuracy. (4) **E5 (Exact Match)** [42]: A dense retrieval method using exact embedding matching without approximation, ensuring the highest retrieval accuracy. The retrieval performance of these methods follows the ranking: E5 (Exact) > E5 (HNSW) > BM25 > Random, as shown in Figure 3(a). Detailed experimental settings and case studies can be found in Appendix G.5 and H.

Results. The training reward curves and final test performance under different search engine settings are presented in Figure 3(c) and Table 4. We observe the following trends: (1) Training with stronger search engines (e.g., **E5 (Exact)** and **E5 (HNSW)**) results in more stable RL training and better final performance. (2) Training with weaker search engines (e.g., **Random** and **BM25**) leads to suboptimal final performance. The search engine call frequency during training is illustrated in

Table 5: Retriever generalization results across datasets and test retrievers. (Qwen2.5-7B-Base, PPO)

Train / Test Search Engine	BM25			E5 (HNSW)			E5 (Exact)			Google Search		
	Bamgb	GPQA	SimpleQA	Bamgb	GPQA	SimpleQA	Bamgb	GPQA	SimpleQA	Bamgb	GPQA	SimpleQA
BM25	0.280	0.273	0.243	0.432	0.293	0.159	0.424	0.323	0.259	0.496	0.313	0.540
E5 (HNSW)	0.240	0.298	0.270	0.400	0.288	0.169	0.440	0.273	0.254	0.528	0.333	0.603
E5 (Exact)	0.312	0.313	0.249	0.400	0.298	0.196	0.424	0.288	0.265	0.560	0.293	0.603
Average	0.277	0.295	0.254	0.411	0.293	0.175	0.429	0.295	0.259	0.528	0.313	0.582

Figure 3(d), revealing: (1) With **Random Noise**, the agent quickly learns to avoid using the search engine, as the retrieved information does not contribute to problem-solving. (2) With **BM25**, the agent gradually increases the number of search engine calls. Since BM25 offers limited retrieval quality, the agent needs to issue multiple, refined queries to obtain relevant information. (3) With **E5** (a stronger retriever), the agent learns to utilize the search engine judiciously, making a reasonable number of calls to acquire the necessary information efficiently.

6.2 Inference with Different Search Engines.

Motivation. In practical scenarios, certain search engines may be unsuitable for integration during RL training due to empirical constraints such as accessibility, cost, or API limitations. In such cases, it becomes necessary to train the agent using one search engine while employing a different one during inference. This raises important research questions: (1) To what extent does the difference in search engines between training and inference affect model performance? (2) Does utilizing a stronger search engine at inference lead to improved downstream performance?

Experimental Design. Following the training setup in [13], we investigate these questions by training the LLM-based search agent using three different search engines: (1) BM25 [33], (2) E5 [42] with approximate nearest neighbor (ANN) search implemented via HNSW [27], and (3) E5 with exact search. During inference, we additionally include the online Google Search API¹ as a stronger retrieval system. To study the importance of the search engine in challenging scenarios, we construct a combined benchmark consisting of 512 samples. This benchmark includes the full Bamboogle test set [30] (Bamgb), the complete GPQA-diamond dataset [32] (GPQA), and a randomly sampled subset from SimpleQA [46]. Detailed experimental settings can be found in Appendix G.6.

Results. As shown in Table 5, we observe the following: (1) LLM search agents trained with a specific search engine demonstrate strong generalization capabilities when evaluated with different search engines during inference. (2) Leveraging a more powerful search engine at inference time (e.g., Google Search) consistently and significantly leads to improved performance, highlighting the importance of high-quality retrieval in downstream applications.

More studies on long-form generation tasks with outcome drive RL and data scaling study can be found in Appendix D and E, respectively.

7 Conclusion

In this work, we conduct comprehensive empirical studies on key design factors in training LLM-based search agents using reinforcement learning. Our investigation reveals that format rewards play an important role in certain scenarios, while intermediate retrieval rewards provide limited benefit and may not consistently improve the learning process. We demonstrate that the choice of the underlying LLM (whether a general-purpose model or one specialized for reasoning) and its scale significantly affect the final agent’s performance. Additionally, the selection of the search engine plays a non-trivial role in shaping both the RL training dynamics and the robustness of the agent during inference. These insights offer practical guidance for developing more capable and reliable LLM-based search agents, paving the way for their deployment in real-world applications. Interesting future directions include exploring more advanced reward modeling techniques, such as learned reward functions and preference-based feedback, as well as studying the agentic behaviors acquired through RL in broader scenarios, including tool use and software engineering.

¹<https://developers.google.com/custom-search/v1/overview>

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2
- [2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024. 2
- [3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 3
- [4] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025. 1, 3
- [5] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019. 16
- [6] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025. 2
- [7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2, 2023. 16
- [8] Carlos Gómez-Rodríguez and Paul Williams. A confederacy of models: A comprehensive evaluation of llms on creative writing. *arXiv preprint arXiv:2310.08433*, 2023. 1
- [9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1, 2, 3, 6, 16
- [10] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 7
- [11] Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag. In *The Thirteenth International Conference on Learning Representations*, 2024. 1, 5
- [12] Bowen Jin, Jinsung Yoon, Zhen Qin, Ziqi Wang, Wei Xiong, Yu Meng, Jiawei Han, and Sercan O Arik. Llm alignment as retriever optimization: An information retrieval perspective. *arXiv preprint arXiv:2502.03699*, 2025. 2
- [13] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. 1, 2, 3, 4, 5, 6, 7, 9, 16, 18, 19
- [14] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 1, 2
- [15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 7
- [16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020. 8
- [17] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 10, 2023. 2

- [18] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. 16
- [19] Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, and Xiaofeng Yang. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025. 1
- [20] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024. 2
- [21] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382, 2023. 2
- [22] Jiacheng Lin, Tian Wang, and Kun Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*, 2025. 2, 5
- [23] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023. 5
- [24] Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*, 2025. 1
- [25] Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, et al. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*, 2025. 1
- [26] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: Surpassing gpt-4 on conversational qa and rag. *Advances in Neural Information Processing Systems*, 37:15416–15459, 2024. 1
- [27] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018. 8, 9
- [28] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2, 3
- [29] Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language model for medical research and healthcare. *NPJ digital medicine*, 6(1):210, 2023. 2
- [30] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022. 7, 9, 20
- [31] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023. 3
- [32] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. 9, 20

- [33] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 2, 8, 9
- [34] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36: 68539–68551, 2023. 1, 3
- [35] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 2
- [36] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025. 3
- [37] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025. 1, 3
- [38] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*, 2022. 16
- [39] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999. 1, 2
- [40] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 2
- [41] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022. 1, 3
- [42] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. 8, 9
- [43] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025. 16
- [44] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. 1
- [45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1
- [46] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024. 9, 20
- [47] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 2
- [48] Xiaobo Xia and Run Luo. Gui-rl: A generalist rl-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025. 1
- [49] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025. 1
- [50] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 6

- 459 [51] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhut-
 460 dinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop
 461 question answering. *arXiv preprint arXiv:1809.09600*, 2018. 16
- 462 [52] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan
 463 Cao. React: Synergizing reasoning and acting in language models. In *International Conference*
 464 *on Learning Representations (ICLR)*, 2023. 3, 4
- 465 [53] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan,
 466 Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning
 467 system at scale. *arXiv preprint arXiv:2503.14476*, 2025. 2
- 468 [54] Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang,
 469 TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement
 470 learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025. 2
- 471 [55] Yufei Zhan, Yousong Zhu, Shurong Zheng, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao
 472 Wang. Vision-rl: Evolving human-free alignment in large vision-language models via vision-
 473 guided reinforcement learning. *arXiv preprint arXiv:2503.18013*, 2025. 1
- 474 [56] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B
 475 Hashimoto. Benchmarking large language models for news summarization. *Transactions of the*
 476 *Association for Computational Linguistics*, 12:39–57, 2024. 1
- 477 [57] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min,
 478 Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv*
 479 *preprint arXiv:2303.18223*, 1(2), 2023. 1, 2
- 480 [58] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and
 481 Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world
 482 environments. *arXiv preprint arXiv:2504.03160*, 2025. 1, 3, 4, 5, 16

A Limitations

In this work, we conduct an empirical investigation into the use of reinforcement learning (RL) for training LLM-based search agents. Our study focuses on three key factors that influence the effectiveness of RL in this context: (1) reward formulation, (2) the choice and characteristics of the underlying language model, and (3) the role and quality of the search engine. However, our analysis is primarily confined to search-based agent scenarios. While this provides valuable insights into the challenges and design choices for RL in retrieval-augmented reasoning, it does not fully generalize to other classes of LLM-based agents. In particular, RL for more complex and open-ended agent behaviors—such as those exhibited by data science assistants, software engineering agents, or multi-tool task planners—remains underexplored. These domains may require more sophisticated reward structures, long-horizon credit assignment, and multi-step decision-making policies that go beyond the scope of search-oriented tasks. We leave a more comprehensive study of RL in broader agentic settings as an important future work direction.

B Positive and Negative Societal Impacts

Our work aims to improve the efficiency and reliability of large language model (LLM)-based agents by leveraging reinforcement learning to optimize their interaction with external tools such as search engines. On the positive side, this line of research can enhance the capabilities of LLMs in high-stakes domains like scientific research, education, and healthcare, where accurate information retrieval and reasoning are critical. By making LLM-based agents more effective at querying and utilizing external knowledge sources, our approach has the potential to reduce hallucinations, improve transparency, and increase user trust in AI systems. However, as with all advances in powerful LLM-based agents, there are potential negative impacts. Improved autonomy in tool usage may lead to unintended misuse, such as generating convincing but misleading information or automating complex tasks without sufficient human oversight. Moreover, the deployment of search-augmented agents could exacerbate access disparities if such technologies are restricted to proprietary systems. Careful consideration of ethical deployment, transparency in agent decision-making, and equitable access to advanced AI capabilities is essential to mitigate these risks.

C Format Reward Code

In this section, we provide the code to judge whether the rollout sequence is in a desired format for a search agent.

Listing 1: Format Reward Code.

```
def is_valid_sequence(text):
    # Find the position of "<|im_start|>assistant" with potential
    # whitespace
    assistant_pattern = r"<|im_start|>assistant\s*"
    assistant_match = re.search(assistant_pattern, text)

    if not assistant_match:
        return False, "Missing assistant marker"

    # Extract the content after the assistant marker
    start_pos = assistant_match.end()
    content = text[start_pos:]

    # Check for balanced tags
    tags_to_check = ["think", "search", "information", "answer"]
    for tag in tags_to_check:
        opening_count = len(re.findall(f"<{tag}>", content))
        closing_count = len(re.findall(f"</{tag}>", content))
        if opening_count != closing_count:
            return False, f"Mismatch in {tag} tags: {opening_count}
                           opening vs {closing_count} closing tags"
```

```

536 # Now check for proper sequence pattern and no extraneous content
537
538 # 1. First split the content by any tags we recognize
539 split_pattern = r"(</?(?:think|search|information|answer)>)"
540 parts = re.split(split_pattern, content)
541
542 # 2. Keep track of the current position in the expected sequence
543 state = "start" # start -> think -> search -> information ->
544 think -> ... -> answer -> end
545
546 # 3. Check each part
547 for i, part in enumerate(parts):
548     # Skip empty parts
549     if not part.strip():
550         continue
551
552     # Check if this is a tag
553     if re.match(r"(</?(?:think|search|information|answer)>)", part):
554         # This is a tag, check if it's valid in the current state
555         if part == "<think>" and state in ["start", "information"]
556             :
557             state = "in_think"
558         elif part == "</think>" and state == "in_think":
559             state = "after_think"
560         elif part == "<search>" and state == "after_think":
561             state = "in_search"
562         elif part == "</search>" and state == "in_search":
563             state = "after_search"
564         elif part == "<information>" and state == "after_search":
565             state = "in_information"
566         elif part == "</information>" and state == "in_information"
567             :
568             state = "information"
569         elif part == "<answer>" and state == "after_think":
570             state = "in_answer"
571         elif part == "</answer>" and state == "in_answer":
572             state = "end"
573         else:
574             return False, f"Unexpected tag {part} in state {state}"
575
576     else:
577         # This is content, check if it's valid in the current
578         state
579         if state in ["in_think", "in_search", "in_information", "
580 in_answer"]
581             :
582             # Content is allowed inside tags
583             pass
584         elif state in ["start", "after_think", "after_search", "
585 information"]
586             :
587             # Only whitespace is allowed between tags
588             if part.strip():
589                 return False, f"Unexpected content '{part.strip()
590 }' between tags (state: {state})"
591             else:
592                 return False, f"Unexpected content in state {state}"
593
594 # Check final state
595 if state != "end":
596     return False, f"Incomplete sequence, ended in state {state}"
597
598 return True, "Valid sequence format"

```

D Reward for Long-form Answers

Motivation. It is demonstrated that rule-based outcome rewards are effective for training LLM-based search agents [13, 58]. However, their evaluation primarily focuses on short-form QA tasks, where answer correctness can be reliably measured using exact string matching. In real-world applications, many queries require long-form, open-ended answers, where evaluating correctness is inherently more subjective. We investigate whether rule-based outcome rewards remain effective in training LLM search agents for long-form QA tasks.

Experimental Design. We conduct experiments on two long-form QA datasets: ASQA [38] and ELI5 [5]. Models are trained on the ASQA training set and evaluated on its development set for in-distribution performance. Out-of-distribution performance is evaluated on the ELI5 dataset. Following common practice, we use the F1 score as the rule-based evaluation metric for both training and evaluation. Experiments are conducted on both Qwen2.5-3B-Base and Qwen2.5-7B-Base models. We compare against several baselines, including Direct Inference (with instruct LLMs), RAG [7] (with instruct LLMs), and R1 [9]. Notably, R1 represents an LLM, trained using the RL approach from DeepSeek-R1 with the same training data (ASQA training set). We also explore a variant of [13, 58] that incorporates the format reward, denoted as “SEARCH-R1 w. Outcome + Format reward”.

Table 6: Study on long-form question answering tasks.

Methods	Qwen2.5-3b			Qwen2.5-7b			Qwen2.5-14b		
	ASQA	ELI5	Avg.	ASQA	ELI5	Avg.	ASQA	ELI5	Avg.
Direct	0.251	0.199	0.225	0.303	0.201	0.252	0.289	0.199	0.244
RAG	0.301	0.193	0.247	0.317	0.202	0.259	0.285	0.193	0.239
R1	0.424	0.275	0.350	0.437	0.280	0.358	0.444	0.278	0.361
SEARCH-R1 w. Outcome + Format reward (PPO)	0.480	0.261	0.370	0.471	0.256	0.363	0.442	0.260	0.351
SEARCH-R1 w. Outcome + Format reward (GRPO)	0.492	0.272	0.382	0.504	0.275	0.390	0.501	0.273	0.387

Results. Performance comparisons are shown in Table 6. We observe that “SEARCH-R1 w. Outcome + Format reward” achieves competitive results on long-form QA tasks, demonstrating the effectiveness of rule-based outcome rewards with format rewards even in complex, open-ended scenarios. This suggests that RL guided by the proposed rewards remains a viable strategy for training search-augmented LLMs, even when the evaluation objective shifts from short-form to long-form answer generation.

E Study of Data Scaling

Motivation. Although RL has shown strong potential in training LLM-based search agents [13], the impact of training data size on the RL process remains underexplored. While recent work has shown that extremely small datasets can be sufficient for reasoning-oriented RL with LLMs [43], it is unclear whether similar data efficiency holds in *agentic RL* settings, where models must learn to reason and interact with external tools in an interleaved manner.

Experimental Design. Following [13], we use the training sets from NQ [18] and HotpotQA [51] as our full training set. To investigate the effect of training data size, we construct subsets of varying sizes by randomly sampling k examples from the full dataset, where $k \in 1, 10, 100, 1000, 10000$. We conduct experiments using the Qwen2.5-3B-Base model and evaluate both PPO and GRPO as the underlying reinforcement learning algorithms.

Results. We present the training reward dynamics and the number of search engine calls across varying training data sizes in Figure 4. The final performance of LLM-based search agents trained with different dataset sizes is reported in Table 7. The results reveal several key observations: (1) Increasing the size of the training dataset generally leads to improved performance, particularly on more complex multi-hop question answering tasks such as HotpotQA and 2Wiki. (2) Smaller datasets lead to faster convergence and higher training rewards, which is likely attributable to overfitting. (3) For PPO, training with a single example fails to induce meaningful search behavior, while using 10 examples results in unstable training. In contrast, training with 100 or 1,000 examples enables the

Table 7: Final performance with different size of training data. $\mathcal{D}$ is the training data. The best performance is set in bold. † / * represents in-domain/out-domain datasets. (LLM: Qwen2.5-3B-Base)

Methods	General QA			Multi-Hop QA				Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2wiki [*]	Musique [*]	Bamboogle [*]	
PPO								
$ \mathcal{D} = 1$	0.121	0.339	0.119	0.141	0.211	0.017	0.056	0.143
$ \mathcal{D} = 10$	0.339	0.509	0.379	0.236	0.237	0.048	0.081	0.261
$ \mathcal{D} = 100$	0.372	0.549	0.382	0.262	0.279	0.063	0.161	0.295
$ \mathcal{D} = 1000$	0.431	0.599	0.446	0.348	0.355	0.136	0.298	0.373
$ \mathcal{D} = 10000$	0.430	0.594	0.445	0.369	0.383	0.155	0.315	0.384
GRPO								
$ \mathcal{D} = 1$	0.107	0.287	0.115	0.124	0.202	0.019	0.089	0.134
$ \mathcal{D} = 10$	0.338	0.515	0.361	0.236	0.223	0.043	0.089	0.258
$ \mathcal{D} = 100$	0.367	0.529	0.414	0.265	0.302	0.079	0.194	0.307
$ \mathcal{D} = 1000$	0.421	0.594	0.437	0.363	0.364	0.149	0.315	0.377
$ \mathcal{D} = 10000$	0.435	0.599	0.435	0.365	0.379	0.137	0.306	0.379

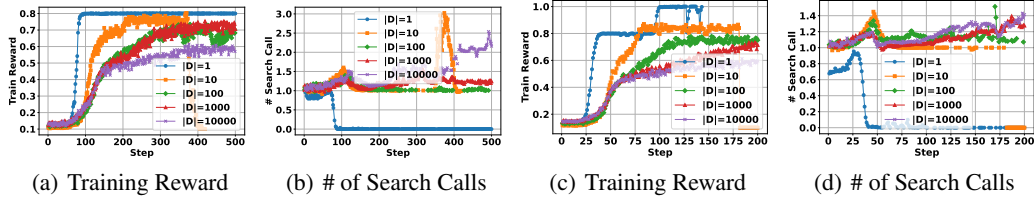


Figure 4: Data scaling effects in RL training for search agents. (a) **Training reward under PPO with varying dataset sizes**: Smaller training sets result in faster convergence and higher training rewards, likely due to overfitting. (b) **Number of search engine calls under PPO**: Training with a single example fails to induce search behavior, while 10 samples lead to unstable learning. In contrast, using 100 or 1,000 samples enables the model to learn stable search behavior, and training with 10,000 samples further improves performance. (c) **Training reward under GRPO with varying dataset sizes**: Similar to PPO, smaller datasets yield faster convergence and higher rewards, again suggesting potential overfitting. (d) **Number of search engine calls under GRPO**: A single training sample is insufficient for search behavior to emerge, whereas larger datasets facilitate stable learning of search interactions.

639 model to learn stable search behavior, and performance continues to improve with 10,000 examples.
640 Similar trends are observed under GRPO training.

641 F More Studies on LLM Types

642 In addition to the 7B model analysis in Section 5.1, we further investigate the impact of LLM
643 initialization on RL performance using 14B-scale models. Specifically, we compare Qwen2.5-14B-
644 Base as a general-purpose LLM and DeepSeek-R1-Distill-Qwen-14B as a reasoning-specialized LLM,
645 evaluating both under PPO and GRPO training. The results, summarized in Table 8 and Figure 5,
646 yield the following observations: (1) RL training is more stable and effective when initialized from the
647 general-purpose LLM, suggesting that such models possess sufficient general reasoning capabilities
648 to support the search agent task, even without reasoning-specific pretraining. (2) Although the
649 reasoning-specialized LLM eventually learns to perform interleaved reasoning and retrieval with PPO
650 and GRPO, the general-purpose LLM consistently achieves higher final performance—likely due to
651 its stronger ability to generate effective search queries.

Table 8: LLM type study with 14B LLMs. The best performance is set in bold. [†]/^{*} represents in-domain/out-domain datasets.

Methods	General QA			Multi-Hop QA				Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2wiki [*]	Musique [*]	Bamboogle [*]	
DeepSeek-R1-Distill-Qwen-14b								
PPO	0.475	0.634	0.465	0.401	0.363	0.211	0.476	0.432
GRPO	0.305	0.613	0.332	0.285	0.276	0.092	0.347	0.321
Qwen2.5-14b-Base								
PPO	0.499	0.680	0.472	0.452	0.431	0.215	0.468	0.459
GRPO	0.500	0.693	0.500	0.481	0.488	0.261	0.516	0.491

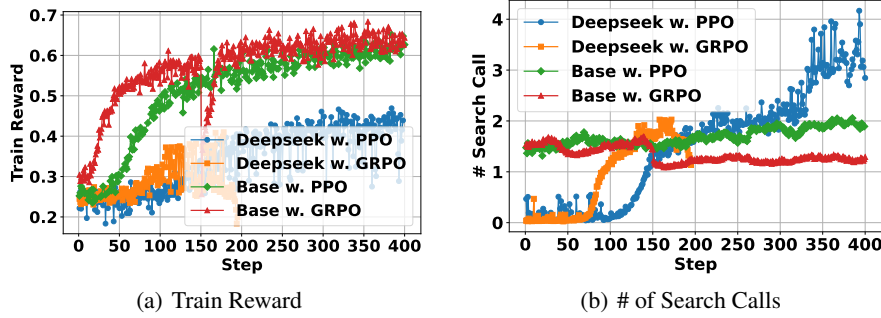


Figure 5: The study of underlying pretrained LLM for development of search agents with RL. (a) **Training reward with different type of LLMs** - general-purpose LLM (Qwen2.5-14B-Base) and reasoning LLM (DeepSeek-R1-Distill-Qwen-14B). We observe that general-purpose LLM performs better than reasoning LLMs with both PPO and GRPO. (b) **# of Search engine calls with different type of LLMs**: Both the general-purpose LLM and the reasoning-specialized LLM demonstrate the ability to learn when to call the search engine. However, the general-purpose LLM achieves better final performance, potentially due to its superior capability in formulating effective search queries.

G Experimental Settings

In this section, we detail the experimental settings used in the studies presented in Sections 4, 5, 6.

G.1 Section 4.1

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. For PPO training, the policy LLM learning rates are set to 1×10^{-6} for Qwen2.5-3B and Qwen2.5-7B, and 5×10^{-7} for Qwen2.5-14B. The critic LLM learning rate is fixed at 1×10^{-5} across all model sizes. For GRPO training, we use a policy LLM learning rate of 5×10^{-7} for all models. The RL training batch size is set to 512, and the rollout temperature is fixed at 1. For GRPO, we set the group size to 5. Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. We conduct a grid search over $\lambda_f \in 0.2, 0.4, 0.6, 0.8$ and select the best-performing value for each model: 0.2 for 3B, 0.4 for 7B, and 0.2 for 14B. We adopt E5 (exact) as the retriever and return the top-3 passages (each contains about 200 tokens).

G.2 Section 4.2

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. For PPO training, the policy LLM learning rates are set to 1×10^{-6} for both Qwen2.5-3B and Qwen2.5-7B. The critic LLM learning rate is fixed at 1×10^{-5} across all model sizes. For GRPO training, we use a policy LLM learning rate of 5×10^{-7} for both models. The RL

training batch size is set to 512, and the rollout temperature is fixed at 1. For GRPO, we set the group size to 5. Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. Based on the findings in Section 4.1, we fix λ_f to 0.2 for the 3B model and 0.4 for the 7B model. We then perform a grid search over $\lambda_r \in 0.1, 0.3, 0.5$ and select the best-performing value for each model, which is 0.1 for both 3B and 7B. We adopt E5 (exact) as the retriever and return the top-3 passages (each contains about 200 tokens).

680 G.3 Section 5.1

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. For PPO training, the policy LLM learning rates are set to 1×10^{-6} for both Qwen2.5-7B and DeepSeek-R1-Distill-Qwen-7B. The critic LLM learning rate is fixed at 1×10^{-5} across all model sizes. For GRPO training, we use a policy LLM learning rate of 5×10^{-7} for both models. The RL training batch size is set to 512, and the rollout temperature is fixed at 1. For GRPO, we set the group size to 5. Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. We set λ_f as 0.2 and λ_r as 0 for all the experiment. We adopt E5 (exact) as the retriever and return the top-3 passages (each contains about 200 tokens).

692 G.4 Section 5.2

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. For GRPO training, we use a policy LLM learning rate of 5×10^{-7} for all models. The RL training batch size is set to 512, and the rollout temperature is fixed at 1. For GRPO, we set the group size to 5. Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. We set λ_f as 0.2 for all the experiment. We adopt E5 (exact) as the retriever and return the top-3 passages (each contains about 200 tokens).

701 G.5 Section 6.1

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. We adopt Qwen2.5-7B-Base as the LLM backbone and PPO as the RL method. For PPO training, the policy LLM learning rates are set to 1×10^{-6} and the critic LLM learning rate is fixed at 1×10^{-5} . Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. For BM25, we adopt the Pyserini implementation² and for E5, we adopt Faiss³. We adopt “HNSW64” for the ANN setting. We return the top-3 passages (each contains about 200 tokens) from each search engine.

711 G.6 Section 6.2

We adopt the same training dataset as used in [13], consisting of the Natural Questions (NQ) and HotpotQA training sets. We adopt Qwen2.5-7B-Base as the LLM backbone and PPO as the RL method. For PPO training, the policy LLM learning rates are set to 1×10^{-6} and the critic LLM learning rate is fixed at 1×10^{-5} . Each model is trained for up to 600 steps, with early stopping triggered if training collapse is observed based on the reward curve. For the results reported in Table 1, we use either the final checkpoint at step 600 or the last checkpoint prior to collapse. All training jobs are conducted on a node equipped with 8 NVIDIA H100 GPUs. For BM25, we adopt the Pyserini implementation and for E5, we adopt Faiss. We adopt “HNSW64” for the ANN setting.

²<https://github.com/castorini/pyserini>

³<https://github.com/facebookresearch/faiss>

720 Given that Google Search API is expensive, we construct a testing set with 512 samples, including
721 full test set from Bamboogle [30] (125 samples), GPQA-Diamond [32] (198 samples) and randomly
722 selected 189 samples from SimpleQA [46]. We return the top-3 passages (each contains about 200
723 tokens) from each search engine.

724 H Case Studies on LLMs Trained with Different Search Engines

Table 9: Case study A of LLM-based search agent trained with different search engines (*i.e.*, random noise, BM25, and E5).

Question: Which Hells Angel member stabbed and beat an attendant of the 1969 counterculture-era rock concert in the United States?
Ground Truth: Alan Passaro
w. random noise:
<code><think></code> To answer this question, we need to recall the incident involving the 1969 counterculture-era rock concert known as Woodstock. The incident in question is the killing of festival security guard Thomas Benson. The Hells Angel member responsible for the stabbing and subsequent death of Benson is CompareTag Stevenson. <code></think></code> <code><answer></code> Manson <code></answer></code>
w. BM25:
<code><think></code> To answer this question, I'll need to identify the Hells Angel member who was involved in the stabbing and beating of an attendant at a 1969 counterculture-era rock concert in the United States. I'll use a search engine to gather information about this event and the responsible Hells Angel member. <code></think></code> <code><search></code> Hells Angel member at 1969 counterculture-era rock concert in the United States <code></search></code> <code><information></code> Doc 1(Title: "Altamont Free Concert") and for the negative portrayal of the Angels in the ""Gimme Shelter"" film. The conspirators reportedly used a boat to approach a residence where Jagger was staying on Long Island, New York, the plot failing when the boat was nearly sunk by a storm. Jagger's spokesperson has refused to comment on the matter ... <code></information></code> <code><think></code> From the search results, I've identified that the 1969 counterculture-era rock concert in question is the Altamont Free Concert. Now, I need to find out which Hells Angel member was involved in stabbing and beating an attendant at this concert. <code></think></code> <code><search></code> Hells Angel member who stabbed and beat an attendant at Altamont Free Concert <code></search></code> <code><information></code> Doc 1(Title: "The Rolling Stones") and former members of the band. Two species, <i>Petroperla mickjaggeri</i> and <i>Lapisperla keithrichardsi</i> , were placed within a new family <i>Petroperlidae</i> . The new family was named in honour of the Rolling Stones, derived from the Greek ""petra"" that stands for ""stone"". The scientists referred to the fossils as ""Rolling Stoneflies"". The Rolling Stones' first concert was on 12 July 1962 at the Marquee Club in London ... <code></information></code> <code><think></code> I have found that a Hells Angel member was responsible for stabbing and beating an attendant at the 1969 Altamont Free Concert. To identify which specific Hells Angel member is being referred to, I will search for the individual who committed this act. <code></think></code> <code><search></code> individual Hells Angel member who stabbed and beat an attendant at Altamont Free Concert <code></search></code> <code><information></code> Doc 1(Title: "The Rolling Stones") and former members of the band. Two species, <i>Petroperla mickjaggeri</i> and <i>Lapisperla keithrichardsi</i> , were placed within a new family <i>Petroperlidae</i> . The new family was named in honour of the Rolling Stones, derived from the Greek ""petra"" that stands for ""stone"". The scientists referred to the fossils as ""Rolling Stoneflies"". The Rolling Stones' first concert was on 12 July 1962 at the Marquee Club in London ... <code></information></code> <code><think></code> After searching for the individual Hells Angel member who stabbed and beat an attendant at the Altamont Free Concert, I have identified that the member was "Ronnie Biggs" <code></think></code> <code><answer></code> Charlie ISHAM <code></answer></code>
w. E5:
<code><think></code> I need to find out which Hells Angel member stabbed and beat an attendant of the 1969 counterculture-era rock concert in the United States. I'll search for it. <code></think></code> <code><search></code> which Hells Angel member stabbed and beat an attendant of the 1969 counterculture-era rock concert in the United States <code></search></code> <code><information></code> Doc 1(Title: "Death of Meredith Hunter") Death of Meredith Hunter Meredith Curly Hunter, Jr. (October 24, 1951 – December 6, 1969) was an 18-year-old African-American teen who was killed at the 1969 Altamont Free Concert. During the performance by The Rolling Stones, Hunter approached the stage, and was violently driven off by members of the Hells ... <code></information></code> <code><think></code> I found out that during the 1969 Altamont Free Concert, a fan named Meredith Hunter was stabbed and beaten to death by Hells Angel Alan Passaro. Now I need to provide the answer. <code></think></code> <code><answer></code> Alan Passaro <code></answer></code>

Table 10: Case B study of LLM-based search agent trained with different search engines (*i.e.*, random noise, BM25, and E5).

Question: Jacques Mathou starred in Delicatessen, the 1991 French movie in the style of which celebrated American animator?
Ground Truth: Terry Gilliam
w. random noise:
<think> To answer this question, we need to consider the animated style of Jacques Mathou’s film "Delicatessen." The movie is known ... American animator. Therefore, the celebrated American animator that "Delicatessen" is in the style of is Tim Burton. </think> <answer> Tim Burton </answer>
w. BM25:
<think> To find out which celebrated American animator Jacques Mathou starred in the 1991 French movie "Delicatessen," I will search for information about the movie and its connections ... </think> <search> Delicatessen 1991 French movie celebrated American animator </search> <information> Doc 1(Title: "Jean-Pierre Jeunet") Jean-Pierre Jeunet Jean-Pierre Jeunet (; born 3 September 1953) is a French film director, producer, and screenwriter. His films are known to mix elements of fantasy, reality and science fiction either to create idealized realities or to give relevance to mundane situations ... </information> <think> From the search results, I have gathered that "Delicatessen" is a 1991 French movie directed by Jean-Pierre Jeunet and Marc Caro. The movie is a post-apocalyptic black comedy. To find the ... </think> <search> Delicatessen celebrated American animator connection </search> <information> Doc 1(Title: Delicatessen) items such as candy and mints are also usually available. Menus vary according to regional ethnic diversity. Although urban delis rely on ethnic meats (such as pastrami, corned beef and salami), supermarket delis rely on meats similar to their packaged meats (primarily ham, turkey and American bologna) ... </information> <think> I have found that the movie "Delicatessen" is associated with Jean-Pierre Jeunet, who is a celebrated film ... for the animator mentioned in the question. </think> <search> Jacques Mathou Delicatessen American animator </search> <information> Doc 1(Title: "Betty Blue") going to the publisher’s house, slashes his face. Zorg induces him to drop charges by threatening him with violence, saying she is the only good thing in his life and she is all he has. Eddy’s mother dies and the friends go to the funeral in Marvejols. There, Eddy asks Zorg and Betty if they will live in the dead woman’s house and look after ... </information> <think> Upon further search, I have found that Jacques Mathou starred in "Delicatessen," but he is not an animator. The question seems to be focused on an American animator related to "Delicatessen." Since I lack this information, .. </think> <answer> Fail to answer </answer>
w. E5:
<think> I need to find out which celebrated American animator Delicatessen, the 1991 French movie, is in the style of. I’ll search for it. </think> <search> Delicatessen, the 1991 French movie in the style of which celebrated .. </search> <information> Doc 1(Title: "Delicatessen (film)") Delicatessen (film) Delicatessen is a 1991 French post-apocalyptic black comedy film directed by Jean-Pierre Jeunet and Marc Caro, starring Dominique Pinon and Karin Viard. It was released in North America as ""presented by Terry Gilliam."" In a dilapidated apartment building in a post-apocalyptic country, food is in short supply ... </information> <think> I found out that Delicatessen is a 1991 French post-apocalyptic black comedy film directed by Jean-Pierre Jeunet and Marc Caro. Now I need to find out which celebrated American animator it is in the style of. </think> <search> which celebrated American animator is Delicatessen in the style of </search> <information> Doc 1(Title: "Joseph Barbera") Joseph Barbera Joseph Roland Barbera (; ; March 24, 1911 – December 18, 2006) was an American animator, director, producer, storyboard artist, and cartoon artist, whose film and television cartoon characters entertained millions of fans worldwide for much of the 20th century. He was born to Italian immigrants in New York City, </information> <think> I found out that Delicatessen is in the style of Terry Gilliam. </think> <answer> Terry Gilliam </answer>

725 In Section 6.1, we have shown that LLMs trained with different search engines will learn different
726 reasoning and search engine calling patterns. In particular, the LLM trained with random noise will
727 learn to not call the search engine since the return from the search engine is not informative; the
728 LLM trained with a weak search engine (*e.g.*, BM25) tend to call the search engine multiple times,
729 while the LLM trained with a stronger search engine (*e.g.*, E5) can call the search engine in a more
730 reasonable pattern.

731 In this section, we would like to show case studies of LLMs trained with different search engine as
732 below in Table 9 and Table 10. The inference time search engine is the same to the training time
733 search engine.

734 From these case studies, we find that during training, higher-quality search engines that provide more
735 relevant information can encourage the agent to achieve its objectives with fewer search calls, as the
736 retrieved content more effectively supports reasoning and decision-making. In contrast, lower-quality
737 search engines that return less relevant information may lead the agent to either over-rely on its
738 internal knowledge or issue multiple search queries to compensate for inadequate results.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: We clearly show in abstract and introduction the contribution and scope of our paper on empirical studies of RL in LLM-based search agent scenarios.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The limitation of this paper is discussed in Appendix A.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the detailed experimental setup to reproduce our empirical studies in Appendix G.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the detailed experimental setup to reproduce our empirical studies in Appendix G. We adopt open-sourced datasets in this paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the detailed experimental setup to reproduce our empirical studies in Appendix G.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Each experiment would take several days or even weeks to complete so it is hard to conduct repeated experiments. However, we ensure consistent random seeds across all methods, thus avoiding randomness in results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the detailed experimental setup to reproduce our empirical studies in Appendix G.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We strictly follow the NeurIPs Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The positive societal impacts and negative societal impacts are discussed in Appendix B

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

998 Answer: [NA]

999 Justification: [NA]

1000 Guidelines:

- 1001 • The answer NA means that the paper poses no such risks.
- 1002 • Released models that have a high risk for misuse or dual-use should be released with
- 1003 necessary safeguards to allow for controlled use of the model, for example by requiring
- 1004 that users adhere to usage guidelines or restrictions to access the model or implementing
- 1005 safety filters.
- 1006 • Datasets that have been scraped from the Internet could pose safety risks. The authors
- 1007 should describe how they avoided releasing unsafe images.
- 1008 • We recognize that providing effective safeguards is challenging, and many papers do
- 1009 not require this, but we encourage authors to take this into account and make a best
- 1010 faith effort.

1011 **12. Licenses for existing assets**

1012 Question: Are the creators or original owners of assets (e.g., code, data, models), used in

1013 the paper, properly credited and are the license and terms of use explicitly mentioned and

1014 properly respected?

1015 Answer: [NA]

1016 Justification: [NA]

1017 Guidelines:

- 1018 • The answer NA means that the paper does not use existing assets.
- 1019 • The authors should cite the original paper that produced the code package or dataset.
- 1020 • The authors should state which version of the asset is used and, if possible, include a
- 1021 URL.
- 1022 • The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- 1023 • For scraped data from a particular source (e.g., website), the copyright and terms of
- 1024 service of that source should be provided.
- 1025 • If assets are released, the license, copyright information, and terms of use in the
- 1026 package should be provided. For popular datasets, paperswithcode.com/datasets
- 1027 has curated licenses for some datasets. Their licensing guide can help determine the
- 1028 license of a dataset.
- 1029 • For existing datasets that are re-packaged, both the original license and the license of
- 1030 the derived asset (if it has changed) should be provided.
- 1031 • If this information is not available online, the authors are encouraged to reach out to
- 1032 the asset's creators.

1033 **13. New assets**

1034 Question: Are new assets introduced in the paper well documented and is the documentation

1035 provided alongside the assets?

1036 Answer: [NA]

1037 Justification: [NA]

1038 Guidelines:

- 1039 • The answer NA means that the paper does not release new assets.
- 1040 • Researchers should communicate the details of the dataset/code/model as part of their
- 1041 submissions via structured templates. This includes details about training, license,
- 1042 limitations, etc.
- 1043 • The paper should discuss whether and how consent was obtained from people whose
- 1044 asset is used.
- 1045 • At submission time, remember to anonymize your assets (if applicable). You can either
- 1046 create an anonymized URL or include an anonymized zip file.

1047 **14. Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This paper focuses on LLM-based search agents, with extensive empirical investigations into training large language models using reinforcement learning.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.