Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning

Bowen Jin[1], **Hansi Zeng**[2], **Zhenrui Yue**[1], **Dong Wang**[1], **Hamed Zamani**[2], **Jiawei Han**[1]
[1] Department of Computer Science, University of Illinois at Urbana-Champaign
[2] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
{bowenj4,zhenrui3,dwang24,hanj}@illinois.edu, {hzeng, zamani}@cs.umass.edu

Abstract

Efficiently acquiring external knowledge and up-to-date information is essential for effective reasoning and text generation in large language models (LLMs). Prompting advanced LLMs with reasoning capabilities during inference to use search engines is not optimal, since the LLM does not learn how to interact optimally with the search engine. This paper introduces SEARCH-R1, an extension of the DeepSeek-R1 model where the LLM learns—solely through reinforcement learning (RL)—to autonomously generate (multiple) search queries during step-by-step reasoning with real-time retrieval. SEARCH-R1 optimizes LLM rollouts with multi-turn search interactions, leveraging retrieved token masking for stable RL training and a simple outcome-based reward function. Experiments on seven question-answering datasets show that SEARCH-R1 improves performance by 26% (Qwen2.5-7B), 21% (Qwen2.5-3B), and 10% (LLaMA3.2-3B) over strong baselines. This paper further provides empirical insights into RL optimization methods, LLM choices, and response length dynamics in retrieval-augmented reasoning. The code and model checkpoints are available at https://github.com/PeterGriffinJin/Search-R1.

1 Introduction

In recent years, large language models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation (Hendrycks et al., 2020; Clark et al., 2018). Despite these achievements, LLMs often encounter challenges when tasked with complex reasoning (Wei et al., 2022) and retrieving up-to-date information from external sources (Jin et al., 2024). Addressing these limitations necessitates integrating advanced reasoning abilities (Huang & Chang, 2022) and the capability to interact effectively with search engines (Schick et al., 2023).

Existing approaches for integrating LLMs with search engines typically fall into two categories: (1) retrieval-augmented generation (RAG) (Gao et al., 2023; Lewis et al., 2020) and (2) treating the search engine as a tool (Yao et al., 2023; Schick et al., 2023). RAG retrieves relevant passages based on the input query and incorporates them into the LLM's context for generation (Lewis et al., 2020). This allows the LLM to leverage external knowledge when answering questions. However, RAG is constrained by retrieval inaccuracy (Jin et al., 2024) and multi-hop retrieval capability (Yang et al., 2018). While existing works (Trivedi et al., 2022a) propose to conduct prompting for multi-turn, multi-query retrieval, it is not optimal, since the LLM does not learn how to interact with the search engine during training. Alternatively, LLMs can be prompted or trained to utilize tools, including search engines, as part of their reasoning process (Qu et al., 2025; Trivedi et al., 2022a). However, prompting-based approaches often struggle with generalization, as certain tasks may not have been encountered during LLM pretraining. On the other hand, training-based approaches provide greater adaptability but rely on large-scale, high-quality annotated trajectories of search-and-reasoning interactions, making them difficult to scale effectively (Schick et al., 2023).

Reinforcement Learning (RL) (Sutton et al., 1999; Kaelbling et al., 1996) has emerged as a potent paradigm for enhancing the reasoning capabilities of LLMs (Guo et al., 2025; Hou et al., 2025; Xie et al., 2025; Kumar et al., 2024). Notably, models like OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025) have leveraged RL techniques (*e.g.*, PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024)) to improve logical inference and problem-solving skills by learning from experience and feedback. After RL, even when trained solely on the outcome rewards, the models learn complex reasoning capabilities, including self-verification (Weng et al., 2022) and self-correction (Kumar et al., 2024).

However, applying reinforcement learning (RL) to search-and-reasoning scenarios presents three key challenges: (1) **RL Framework and Stability** – It remains unclear how to effectively integrate the search engine into the LLM RL framework while ensuring stable optimization, particularly when incorporating retrieved context. (2) **Multi-Turn Interleaved Reasoning and Search** – Ideally, the LLM should be capable of iterative reasoning and search engine calls, dynamically adjusting its retrieval strategy based on the complexity of the problem. (3) **Reward Design** – Designing an effective reward function for search-and-reasoning tasks is nontrivial, as traditional reward formulations may not generalize well to this new paradigm.

To address these challenges, we introduce SEARCH-R1, a novel reinforcement learning (RL) framework that enables LLMs to interact with search engines in an interleaved manner with their own reasoning. Specifically, SEARCH-R1 introduces the following key innovations: (1) We model the search engine as part of the environment, enabling rollout sequences that interleave LLM token generation with search engine retrievals. SEARCH-R1 is compatible with various RL algorithms, including PPO and GRPO, and we apply retrieved token masking to ensure stable optimization. (2) SEARCH-R1 supports multi-turn retrieval and reasoning, where search calls are explicitly triggered by `<search>` and `</search>` tokens. Retrieved content is enclosed within `<information>` and `</information>` tokens, while LLM reasoning steps are wrapped within `<think>` and `</think>` tokens. The final answer is formatted using `<answer>` and `</answer>` tokens, allowing for structured, iterative decision-making. (3) We adopt a straightforward outcome-based reward function, avoiding the complexity of process-based rewards. Our results demonstrate that this minimal reward design is effective in search-and-reasoning scenarios. SEARCH-R1 can be viewed as an extension of DeepSeek-R1 (Guo et al., 2025), which primarily focuses on parametric reasoning by introducing search-augmented RL training for enhanced retrieval-driven decision-making.

In summary, our key contributions are threefold:

- We identify the challenges of applying RL to LLM reasoning with search engine calling.
- We propose SEARCH-R1, a novel reinforcement learning framework that supports LLM rollout and RL optimization with a search engine, including retrieved token masking to stabilize RL training, multi-turn interleaved reasoning and search to support complex task-solving and a simple yet effective outcome reward function.
- We conduct systematic experiments to demonstrate the effectiveness of SEARCH-R1 with 26%, 21%, and 10% average relative improvement with three LLMs over strong baselines. In addition, we provide insights on RL for reasoning and search settings, including RL methods selection, different LLM choices and response length study.

2 Related Works

2.1 Large Language Models and Retrieval

Although large language models (LLMs) (Zhao et al., 2023; Team, 2024; Achiam et al., 2023) have demonstrated remarkable reasoning (Guo et al., 2025) and coding (Guo et al., 2024) capabilities, they still lack domain-specific knowledge (Peng et al., 2023; Li et al., 2023) and are prone to hallucinations (Zhang et al., 2023). To address these limitations, search engines (Zhao et al., 2024) are widely used to provide external information. There are two primary ways to integrate search engines with LLMs: (1) retrieval-augmented generation (RAG) (Gao et al., 2023) and (2) treating the search engine as a tool (Schick et al., 2023). RAG (Lewis et al., 2020; Yue et al., 2024; Xiong et al., 2025) typically follows a one-round retrieval and

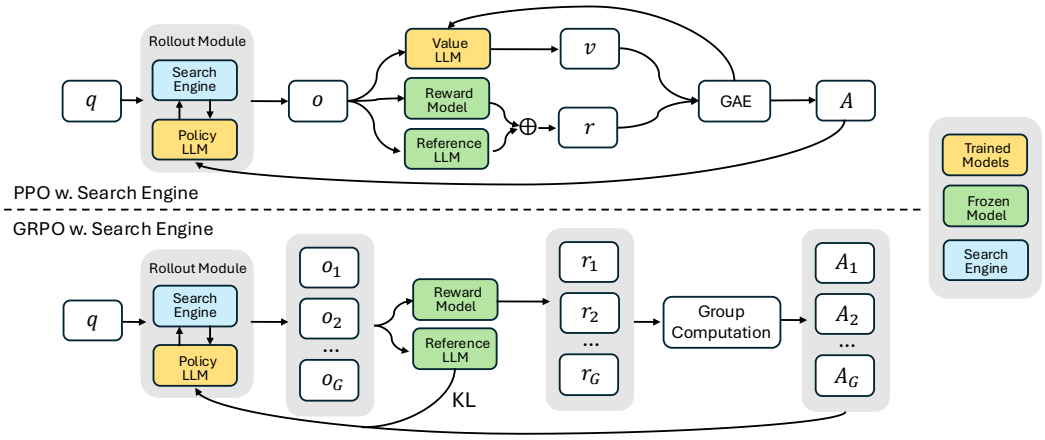


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1).

sequential generation pipeline, where a search engine fetches relevant information based on the input query, which is then concatenated with the query and fed into the LLM. However, this pipeline struggles with issues such as retrieving irrelevant information (Jin et al., 2024) and failing to provide sufficiently useful context (Jiang et al., 2023). An alternative approach is search-as-a-tool, where LLMs are prompted or fine-tuned to interact with the search engine. IRCoT (Trivedi et al., 2022a) and ReAct (Yao et al., 2023) use prompting to guide iterative reasoning and search engine calls, while Toolformer (Schick et al., 2023) leverages supervised fine-tuning to enhance search capabilities. However, these methods rely on high-quality labeled trajectories, which are difficult to scale. Recent work (Guo et al., 2025) suggests that reinforcement learning can enable LLMs to develop advanced reasoning skills using only outcome rewards, yet its potential in search engine calling scenarios remains under-explored.

2.2 Large Language Models and Reinforcement Learning

Reinforcement learning (RL) (Kaelbling et al., 1996) is a learning paradigm where an agent learns to make sequential decisions by interacting with an environment and receiving feedback in the form of rewards, aiming to maximize cumulative reward over time (Sutton et al., 1999). RL was introduced to LLM tuning by Ouyang et al. (2022) through reinforcement learning from human feedback (RLHF) (Kaufmann et al., 2023). This approach first trains a reward model using human preference data (Lambert et al., 2024), which then guides RL-based tuning of the policy LLM, typically via the Proximal Policy Optimization (PPO) algorithm. However, PPO involves multiple rounds of LLM optimization, making it challenging to implement. To simplify RL-based tuning, direct optimization methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and SimPO (Meng et al., 2024) have been proposed. While these methods offer computational efficiency, they suffer from off-policy issues (Pang et al., 2024) and do not consistently match the performance of pure RL approaches. Alternative solutions include Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which eliminates the need for a critic model by estimating baselines from group scores, and RLOO (Ahmadian et al., 2024), which introduces a simplified REINFORCE-style (Williams, 1992) optimization framework. Despite these advances, the application of RL to LLM-driven search engine interactions and reasoning remains largely unexplored.

3 Search-R1

In the following sections, we present the detailed design of SEARCH-R1, covering (1) reinforcement learning with a search engine; (2) text generation with an interleaved multi-turn search engine call; (3) the training template; and (4) reward model design.

3.1 Reinforcement Learning with a Search Engine

We formulate the reinforcement learning framework with a search engine $\mathcal{R}$ as follows:

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

where π_θ is the policy LLM, π_{ref} is the reference LLM, r_ϕ is the reward function and $\mathbb{D}_{\mathrm{KL}}$ is the KL-divergence. Unlike prior LLM reinforcement learning methods that primarily rely on the policy LLM $\pi_\theta(\cdot \mid x)$ to generate rollout sequences (Rafailov et al., 2023; Ouyang et al., 2022), our framework explicitly incorporates retrieval interleaved reasoning via $\pi_\theta(\cdot \mid x;\mathcal{R})$, which can be seen as $\pi_\theta(\cdot \mid x) \bigotimes \mathcal{R}$, where $\bigotimes$ denotes interleaved retrieval-and-reasoning. This enables more effective decision-making in reasoning-intensive tasks that require external information retrieval. A detailed illustration of the rollout process is provided in Section 3.2.

Our approach builds upon two well-established policy gradient RL methods: Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025), leveraging their respective advantages to optimize retrieval-augmented reasoning.

Loss Masking for Retrieved Tokens. In both PPO and GRPO, the token-level loss is computed over the entire rollout sequence. In SEARCH-R1, the rollout sequence consists of both LLM-generated tokens and retrieved tokens from external passages. While optimizing LLM-generated tokens enhances the model's ability to interact with the search engine and perform reasoning, applying the same optimization to retrieved tokens can lead to unintended learning dynamics. To address this, we introduce loss masking for retrieved tokens, ensuring that the policy gradient objective is computed only over LLM-generated tokens, while excluding retrieved content from the optimization process. This approach stabilizes training while preserving the flexibility of search-augmented generation.

PPO + Search Engine. Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a popular actor-critic reinforcement learning algorithm commonly used for fine-tuning large language models (LLMs) during the RL stage (Ouyang et al., 2022). In our reasoning plus search engine calling scenario, it optimizes LLMs by maximizing the following objective:

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\mathrm{old}}(\cdot|x;\mathcal{R})}\left[\frac{1}{\sum_{t=1}^{|y|} I(y_t)} \sum_{t=1:I(y_t)=1}^{|y|} \min\left(\frac{\pi_\theta(y_t|x,y_{<t};\mathcal{R})}{\pi_{\mathrm{old}}(y_t|x,y_{<t};\mathcal{R})} A_t, \mathrm{clip}\left(\frac{\pi_\theta(y_t|x,y_{<t};\mathcal{R})}{\pi_{\mathrm{old}}(y_t|x,y_{<t};\mathcal{R})}, 1-\epsilon, 1+\epsilon\right) A_t\right)\right], \quad (1)$$

where π_θ and π_{ref} represent the current and reference policy models, respectively. The variable x denotes input samples drawn from the dataset $\mathcal{D}$, while y represents the model's generated outputs interleaved with search engine calling results, sampled from the reference policy $\pi_{\mathrm{ref}}(y \mid x;\mathcal{R})$ and retrieved from the search engine $\mathcal{R}$. $I(y_t)$ is the token loss masking operation. $I(y_t) = 1$ if y_t is a LLM generated token and $I(y_t) = 0$ if y_t is a retrieved token. The term ϵ is a clipping-related hyperparameter introduced in PPO to stabilize training. The advantage estimate A_t is computed using Generalized Advantage Estimation (GAE) (Schulman et al., 2015), based on future rewards $\{r_{\geq t}\}$ and a learned value function V_ϕ.

GRPO + Search Engine. To improve policy optimization stability and avoid the need for an additional value function approximation, Group Relative Policy Optimization (GRPO) is introduced in Shao et al. (2024). GRPO differs from Proximal Policy Optimization (PPO) by leveraging the average reward of multiple sampled outputs as a baseline rather than relying on a learned value function. Specifically, for each input question x, GRPO samples a group of responses $\{y_1, y_2, \ldots, y_G\}$ from the reference policy π_{ref}. The policy model is then optimized by maximizing the following objective function:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|x;\mathcal{R})} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{\sum_{t=1}^{|y_i|} I(y_{i,t})} \sum_{t=1:I(y_{i,t})=1}^{|y_i|} \min \left(\frac{\pi_\theta(y_{i,t}|x, y_{i,<t};\mathcal{R})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t};\mathcal{R})} \hat{A}_{i,t}, \right. \right.$$
$$\left. \left. \text{clip}\left(\frac{\pi_\theta(y_{i,t}|x, y_{i,<t};\mathcal{R})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t};\mathcal{R})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \right) - \beta \mathbb{D}_{KL}\left[\pi_\theta || \pi_{\text{ref}}\right] \right], \quad (2)$$

where ϵ and β are hyperparameters, and $\hat{A}_{i,t}$ represents the advantage, which is computed based on the relative rewards of outputs within each group. This approach avoids introducing additional complexity in the computation of $\hat{A}_{i,t}$. $I(y_{i,t})$ is the token loss masking operation. $I(y_{i,t}) = 1$ if $y_{i,t}$ is a LLM generated token and $I(y_{i,t}) = 0$ if $y_{i,t}$ is a retrieved token. Additionally, instead of incorporating KL divergence as a penalty within the reward function, GRPO regularizes by directly adding the KL divergence between the trained policy and the reference policy to the loss function. The retrieved token masking is also applied when calculating the KL divergence loss $\mathbb{D}_{KL}$.

3.2 Text Generation with Interleaved Multi-turn Search Engine Call

In this section, we describe the rollout process for LLM response generation with interleaved multi-turn search engine calls, formulated as: $y \sim \pi_\theta(\cdot \mid x; \mathcal{R}) = \pi_\theta(\cdot \mid x) \bigotimes \mathcal{R}$.

Our approach follows an iterative framework where the LLM alternates between text generation and external search engine queries. Specifically, the system instruction guides the LLM to encapsulate its search query between two designated search call tokens, <search> and </search>, whenever an external retrieval is needed. Upon detecting these tokens in the generated sequence, the system extracts the search query, queries the search engine, and retrieves relevant results. The retrieved information is then enclosed within special retrieval tokens, <information> and </information>, and appended to the ongoing rollout sequence, serving as additional context for the next generation step. This process continues iteratively until one of the following conditions is met: (1) the search engine call budget is exhausted, or (2) the model generates a final response, which is enclosed between designated answer tokens, <answer> and </answer>. The complete workflow is outlined in Algorithm 1.

3.3 Training Template

To train SEARCH-R1, we start by crafting a simple template that directs the initial LLM to follow our predefined instructions. As shown in Table 1, this template structures the model's output into three parts in an iterative fashion: first, a reasoning process, then a search engine calling function, and finally, the answer. We deliberately limit our constraints to this structural format, avoiding any content-specific biases, such as enforcing reflective reasoning and search engine calling or endorsing specific problem-solving approaches. This ensures that the model's natural learning dynamics during the RL process remain observable and unbiased.

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search>, and it will return the top searched results between <information> and </information>. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer> without detailed illustrations. For example, <answer> Beijing </answer>. Question: question.

Table 1: Template for SEARCH-R1. question will be replaced with the specific question during training and inference.

Algorithm 1 LLM Response Rollout with Multi-Turn Search Engine Calls

Require: Input query x, policy model π_θ, search engine $\mathcal{R}$, maximum search budget B.
Ensure: Final response y.

Initialize rollout sequence $y \leftarrow \emptyset$
Initialize search call count $b \leftarrow 0$
while $b < B$ **do**
  Generate response token $y_t \sim \pi_\theta(\cdot \mid x, y)$
  // Append y_t to rollout sequence y
  $y \leftarrow y + y_t$
  if <search> </search> detected in y_t **then**
    // Extract search query q
    $q \leftarrow \text{Parse}(y_t, \texttt{<search>}, \texttt{</search>})$
    // Retrieve search results
    $d = \mathcal{R}(q)$
    // Insert d into y
    $y \leftarrow y + \texttt{<information>}d\texttt{</information>}$
    Increment search call count $b \leftarrow b + 1$
  end if
  if <answer> </answer> detected in y **then**
    // Terminate rollout
    return final generated response y
  end if
end while
return final generated response y

3.4 Reward Modeling

The reward function serves as the primary training signal, guiding the optimization process in reinforcement learning. To train SEARCH-R1, we adopt a rule-based reward system that consists solely of **final outcome rewards**, which assess the correctness of the model's response. For instance, in factual reasoning tasks, correctness can be evaluated using rule-based criteria such as exact string matching.

$$r_\phi(x, y) = \text{EM}(a_{\text{pred}}, a_{\text{gold}}), \tag{3}$$

where a_{pred} is the extracted final answer from response y and a_{gold} is the ground truth answer. Unlike Guo et al. (2025), we do not incorporate format rewards, as our learned model already demonstrates strong structural adherence. We leave the exploration of more complex format rewards for future work. Furthermore, we deliberately avoid training neural reward models for either outcome or process evaluation, following Guo et al. (2025). This decision is motivated by the susceptibility of neural reward models to reward hacking in large-scale reinforcement learning, as well as the additional computational cost and complexity introduced by retraining these models.

4 Main results

4.1 Datasets

We evaluate SEARCH-R1 on seven benchmark datasets, categorized as follows:

General Question Answering: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2022).

Multi-Hop Question Answering: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), Musique (Trivedi et al., 2022b), and Bamboogle (Press et al., 2022).

These datasets encompass a diverse range of search with reasoning challenges, enabling a comprehensive evaluation of SEARCH-R1 across both single-turn and multi-hop retrieval scenarios.

4.2 Baselines

To evaluate the effectiveness of SEARCH-R1, we compare it against the following baseline methods:

Inference without Retrieval: Direct inference and Chain-of-Thought (CoT) reasoning (Wei et al., 2022).

Inference with Retrieval: Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), IRCoT (Trivedi et al., 2022a), and Search-o1 (Li et al., 2025).

Fine-Tuning-Based Methods: Supervised fine-tuning (SFT) (Chung et al., 2024) and reinforcement learning-based fine-tuning without a search engine (R1) (Guo et al., 2025). For R1, we train the LLMs with the RL methods proposed in Guo et al. (2025) with our data to have a fair comparison with SEARCH-R1. It only contains reasoning and answer steps and cannot call a search engine.

These baselines cover a broad spectrum of retrieval-augmented and fine-tuning approaches, allowing for a comprehensive assessment of SEARCH-R1 in both zero-shot and learned retrieval settings.

To make a fair comparison between different methods, we use the same retriever, knowledge corpus, training data and LLMs. More details can be found in Section 4.3.

4.3 Experimental Setup

We conduct experiments using three types of models: Qwen-2.5-3B (Base/Instruct) and Qwen-2.5-7B (Base/Instruct) (Yang et al., 2024), as well as Llama-3.2-3B (Base/Instruct) (Dubey et al., 2024). For retrieval, we use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2022) as the retriever. To ensure fair comparison, we follow Lin et al. (2023) and set the number of retrieved passages to three across all retrieval-based methods.

For training, we merge the training sets of NQ and HotpotQA to form a unified dataset for SEARCH-R1 and other fine-tuning-based baselines. Evaluation is conducted on the test or validation sets of all seven datasets to assess both in-domain and out-of-domain performance. Exact Match (EM) is used as the evaluation metric, following Yu et al. (2024). For inference-style baselines, we use instruct models, as base models fail to follow instructions. For RL tuning methods, experiments are conducted on both base and instruct models.

For SEARCH-R1 training, in PPO Training, the policy LLM learning rate is set to 1e-6, and value LLM learning rate to 1e-5. The Generalized Advantage Estimation (GAE) parameters are $\lambda = 1$ and $\gamma = 1$. In GRPO Training, the policy LLM learning rate is set to 1e-6, with five sampled responses per prompt. We use exact match (EM) to calculate the outcome reward. Unless stated otherwise, **PPO is used as the default RL method**, and a detailed comparison between PPO and GRPO is provided in Section 5.1.

4.4 Performance

The main results comparing SEARCH-R1 with baseline methods across the seven datasets are presented in Table 2. From the results, we make the following key observations:

SEARCH-R1 consistently outperforms strong baseline methods. We achieve 26%, 21%, and 10% average relative improvement with Qwen2.5-7B, Qwen2.5-3B, and LLaMA3.2-3B, respectively. These gains hold across both in-distribution evaluation (*i.e.*, NQ and HotpotQA) and out-of-distribution evaluation (*i.e.*, TriviaQA, PopQA, 2WikiMultiHopQA, Musique, and Bamboogle).

SEARCH-R1 surpasses RL-based training for LLM reasoning without retrieval (R1) (Guo et al., 2025). This aligns with expectations, as incorporating search into LLM reasoning provides access to relevant external knowledge, improving overall performance.

Table 2: Main results. The best performance is set in bold, and the second best is set in underline.

Method	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	Avg.
Qwen2.5-7b-Base/Instruct								
Direct Inference	0.134	0.408	0.140	0.183	0.250	0.031	0.120	0.181
CoT	0.048	0.185	0.054	0.092	0.111	0.022	0.232	0.106
IRCoT	0.224	0.478	0.301	0.133	0.149	0.072	0.224	0.239
Search-o1	0.151	0.443	0.131	0.187	0.176	0.058	0.296	0.206
RAG	0.349	0.585	0.392	0.299	0.235	0.058	0.208	0.304
SFT	0.318	0.354	0.121	0.217	0.259	0.066	0.112	0.207
R1-base	0.297	0.539	0.202	0.242	0.273	0.083	0.296	0.276
R1-instruct	0.270	0.537	0.199	0.237	0.292	0.072	0.293	0.271
Search-R1-base	**0.412**	0.568	**0.428**	0.356	0.322	0.142	0.384	0.373
Search-R1-instruct	0.397	**0.606**	0.404	**0.380**	**0.326**	**0.168**	**0.408**	**0.384**
Qwen2.5-3b-Base/Instruct								
Direct Inference	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
CoT	0.023	0.032	0.005	0.021	0.021	0.002	0.000	0.015
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	**0.320**	0.255
RAG	**0.348**	**0.544**	**0.387**	0.255	0.226	0.047	0.080	0.270
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
R1-instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Search-R1-base	0.341	0.513	0.362	0.263	0.273	0.076	0.211	0.292
Search-R1-instruct	0.323	0.537	0.364	**0.308**	**0.336**	**0.105**	0.315	**0.327**
LLaMA3.2-3b-Base/Instruct								
Direct Inference	0.139	0.368	0.124	0.122	0.107	0.015	0.064	0.134
CoT	0.246	0.487	0.166	0.051	0.083	0.006	0.024	0.152
IRCoT	0.363	0.566	0.428	0.238	0.236	0.072	0.208	0.301
Search-o1	0.107	0.203	0.093	0.132	0.117	0.035	0.176	0.123
RAG	0.317	0.551	0.337	0.234	0.118	0.034	0.064	0.237
SFT	0.320	0.341	0.122	0.206	0.257	0.064	0.120	0.204
R1-base	0.290	0.514	0.237	0.234	**0.279**	0.055	0.146	0.251
R1-instruct	0.384	0.549	0.228	0.238	0.269	0.074	**0.315**	0.294
Search-R1-base	**0.394**	**0.596**	**0.437**	0.280	0.264	0.056	0.105	0.305
Search-R1-instruct	0.357	0.578	0.378	**0.314**	0.233	**0.090**	0.306	**0.322**

SEARCH-R1 is effective for both base and instruction-tuned models. This demonstrates that DeepSeek-R1-Zero-style RL with outcome-based rewards (Guo et al., 2025) can be successfully applied to reasoning with search, extending beyond its previously established effectiveness in pure reasoning scenarios.

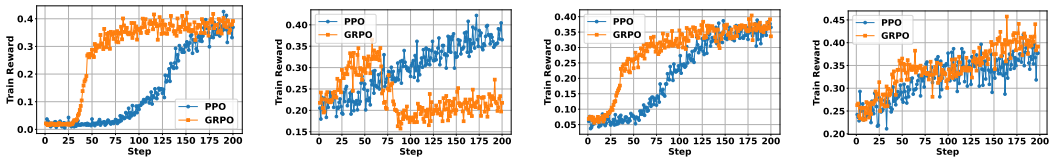
SEARCH-R1 generalizes across different base LLMs, including Qwen2.5 and LLaMA3.2. This contrasts with findings in RL for mathematical reasoning, where RL has been observed to work effectively only for certain base LLMs (Zeng et al., 2025). Our results indicate that search-augmented RL is more broadly applicable across model families.

5 Analysis

5.1 Different RL methods: PPO vs. GRPO

We evaluate SEARCH-R1 using both PPO and GRPO as the base RL method, conducting experiments on LLaMA3.2-3B and Qwen2.5-3B models. The training dynamics comparison is presented in Figure 2, revealing the following insights:

GRPO converges faster than PPO across all cases. This is because PPO relies on a critic model, which requires several warm-up steps before effective training begins.



(a) LLaMA3.2-3b-base (b) LLaMA3.2-3b-it (c) Qwen2.5-3b-base (d) Qwen2.5-3b-it

Figure 2: Training dynamics of SEARCH-R1 with PPO and GRPO as the base RL method across four LLMs. GRPO generally converges faster but may exhibit instability in certain cases (*e.g.*, LLaMA3.2-3B-Instruct), whereas PPO provides more stable optimization but converges at a slower rate.

Table 3: The performance results of SEARCH-R1 with PPO and GRPO on seven datasets.

Method	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	Avg.
Qwen2.5-3b-Base/Instruct								
SEARCH-R1-base (GRPO)	0.396	**0.582**	0.390	0.283	0.266	0.054	0.113	0.298
SEARCH-R1-instruct (GRPO)	**0.409**	0.552	**0.405**	**0.345**	**0.369**	**0.154**	**0.320**	**0.365**
SEARCH-R1-base (PPO)	0.341	0.513	0.362	0.263	0.273	0.076	0.211	0.292
SEARCH-R1-instruct (PPO)	0.323	0.537	0.364	0.308	0.336	0.105	0.315	0.327
LLaMA3.2-3b-Base/Instruct								
SEARCH-R1-base (GRPO)	**0.431**	**0.612**	**0.458**	0.300	**0.297**	0.067	0.104	**0.324**
SEARCH-R1-instruct (GRPO)	0.333	0.524	0.329	0.229	0.190	0.047	0.192	0.263
SEARCH-R1-base (PPO)	0.394	0.596	0.437	0.280	0.264	0.056	0.105	0.305
SEARCH-R1-instruct (PPO)	0.357	0.578	0.378	**0.314**	0.233	**0.090**	**0.306**	0.322

PPO demonstrates greater training stability. As shown in Figure 2(b), GRPO leads to reward collapse when applied to the LLaMA3.2-3B-Instruct model, whereas PPO remains stable across different LLM architectures.

The final training rewards of PPO and GRPO are comparable. Despite differences in convergence speed and stability, both methods achieve similar final reward values, indicating that both are viable for optimizing SEARCH-R1.

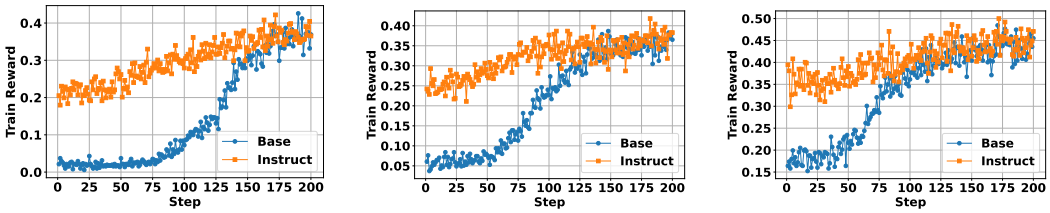
The evaluation results are presented in Table 3, revealing the following key findings:

GRPO generally outperforms PPO. Across both Qwen2.5-3B and LLaMA3.2-3B, GRPO achieves higher average performance, demonstrating its effectiveness in optimizing retrieval-augmented reasoning.

Instruct variants perform better than base variants. For Qwen2.5-3B, SEARCH-R1-Instruct (GRPO) achieves the highest overall average score (0.365), outperforming all other configurations. For LLaMA3.2-3B, the best-performing variant is SEARCH-R1-Base (GRPO) with an average score of 0.324, followed closely by SEARCH-R1-Instruct (PPO) at 0.322.

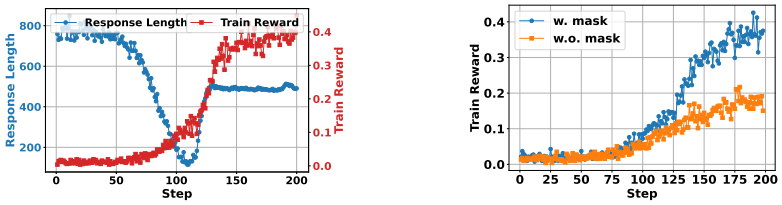
5.2 Base vs. Instruct LLMs

We analyze the training dynamics of SEARCH-R1 across both base LLMs and instruction-tuned LLMs. Experiments are conducted on three model variants: LLaMA3.2-3B, Qwen2.5-3B, and Qwen2.5-7B. As shown in Figure 3, we observe that instruction-tuned models converge faster and start from a higher initial performance compared to base models. However, the final performance of both model types remains highly similar after training. This finding suggests that while general post-training accelerates learning in reasoning-plus-search scenarios, reinforcement learning can effectively bridge the gap over time, enabling base models to achieve comparable performance.



(a) LLaMA3.2-3b-base/instruct (b) Qwen2.5-3b-base/instruct (c) Qwen2.5-7b-base/instruct

Figure 3: Study of SEARCH-R1 on base and instruct LLMs. The instruction model converges faster and starts from a better initial performance. However, the final performance of both models is very similar.



(a) Response length study (b) Loss mask dynamics study

Figure 4: (a) Response Length Study: The response length exhibits a decrease-increase-stabilize trend throughout training, aligning with the overall performance trajectory of the LLM. (b) Retrieved Token Loss Masking Study: Implementing retrieved token masking leads to greater LLM improvements, mitigating unintended optimization effects and ensuring more stable training dynamics.

5.3 Response Length Study

We conduct an experiment using SEARCH-R1 with the LLaMA3.2-3b-base model, training on NQ to analyze the dynamics of training reward and response length over the course of training. The results are presented in Figure 4(a), revealing the following key trends:

(1) Early Stage (First 100 Steps): The response length sharply decreases, while the training reward exhibits a slight increase. During this phase, the base model learns to eliminate excessive filler words and begins adapting to the task requirements.

(2) Mid Stage (100–130 Steps): Both response length and training reward increase significantly. At this point, the LLM learns to call the search engine, resulting in longer responses due to retrieved passages. The training reward improves substantially, as the model becomes more effective at leveraging search results.

(3) Late Stage (After 130 Steps): The response length stabilizes, and the training reward continues to increase slightly. At this stage, the model has learned to use the search engine effectively and focuses on refining its search queries. Given that NQ is a relatively simple task, the response length stabilizes at approximately 500 tokens, indicating convergence.

5.4 Study of Retrieved Tokens Loss Masking

In Section 3.1, we introduced token-level loss masking for retrieved tokens to prevent unintended optimization behaviors. Here, we empirically evaluate its effectiveness by analyzing its impact on training stability and model performance.

We conduct experiments on the LLaMA3.2-3b-base model, comparing training dynamics with and without retrieved token loss masking. As shown in Figure 4(b), applying retrieved token masking results in greater LLM improvements, mitigating unintended optimization effects and ensuring more stable training dynamics.

Table 4: The performance of SEARCH-R1 with and without retrieved token loss masking. The LLM trained with retrieved token loss masking achieves consistently better performance. (LLM: LLaMA3.2-3b-base)

Method	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	Avg.
SEARCH-R1 w. mask	**0.394**	**0.596**	**0.437**	**0.280**	**0.264**	**0.056**	**0.105**	**0.305**
SEARCH-R1 w.o. mask	0.124	0.360	0.152	0.131	0.211	0.017	0.032	0.147

Table 5: A case study of R1 and SEARCH-R1.

Question: Curious is a women's fragrance by a singer born in what city and state?

Ground Truth: McComb, Mississippi

R1:

<think> The singer is Beyoncé, who was born in Houston, Texas. </think>
<answer> Houston </answer>

SEARCH-R1:

<think> I need to find out which city and state a singer, by whom the women's fragrance "Curious" was created, was born in. </think>
<search> Curious fragrance information </search>
<information> Doc 1(Title: "Curious (fragrance)") Curious (fragrance) Curious is a women's fragrance by Britney Spears for Elizabeth Arden, and is the first perfume to be endorsed by Britney Spears, preceding ""Fantasy."" Spears reportedly earned $52 million from the endorsement. ""Curious"" was released in September 2004, and was very successful internationally. It was the number one fragrance of 2004 in department stores, and in 2005, ""Curious"" was honored by the Fragrance Foundation as Best Women's Fragrance. The fragrance is a white floral scent. Its notes are Louisiana magnolia, golden Anjou pear, lotus flower, tuberose, star jasmine, pink cyclamen, vanilla-infused musk, sandalwood and blonde woods. Elizabeth Arden ... </information>
<think> I need to find out which city and state Britney Spears, the singer behind "Curious" fragrance, was born in. </think>
<search> Britney Spears birthplace </search>
<information> Doc 1(Title: "Britney Spears") the best-selling female albums artist of the 2000s. In 2004, she launched a perfume brand with Elizabeth Arden, Inc., from which sales exceeded US$1.5 billion, . Spears was born in McComb, Mississippi, the second child of Lynne Irene Bridges and James Parnell Spears. Her maternal grandmother, Lillian Portell, was English (born in London), and one of Spears's maternal great-great-grandfathers was Maltese. Her siblings are Bryan James and Jamie Lynn. Britney was born in the Bible Belt, where socially conservative evangelical Protestantism is a particularly strong religious influence. Spears was baptized into the Southern Baptist Convention, but later in life studied </information>
<think> Since Britney Spears was born in McComb, Mississippi, I need to find out which city and state that is. </think>
<search> McComb, Mississippi location </search>
<information> Doc 1(Title: "McComb, Mississippi") McComb, Mississippi McComb is a city in Pike County, Mississippi, United States, approximately south of Jackson. As of the 2010 census, the city had a total population of 12,790. It is the principal city of the McComb, Mississippi Micropolitan Statistical Area. McComb was founded in 1872 after Henry Simpson McComb of the New Orleans, Jackson and Great Northern Railroad, a predecessor of the Illinois Central Railroad (now part of the Canadian National Railway), decided to move the railroad's maintenance shops away from New Orleans, Louisiana, to avoid the attractions of that city's saloons. The railroad purchased land in Pike County, </information>
<think> I have found that Britney Spears was born in McComb, Mississippi. Since the question asks for the city and state, I just need to confirm the state, which I already know from the previous search. </think>
<answer> McComb, Mississippi </answer>

The performance comparison is provided in Table 4, demonstrating that SEARCH-R1 trained with retrieved token loss masking consistently outperforms the variant without masking.

5.5 Case Studies

To gain deeper insights into SEARCH-R1, we conduct a case study using Qwen2.5-7B-Base, comparing its behavior with RL without a search engine (Guo et al., 2025). The results are presented in Table 5, revealing the following key observations:

Interleaved Reasoning and Retrieval Enhances Problem Analysis: SEARCH-R1 enables the LLM to perform in-depth reasoning with multi-turn retrieval, whereas RL without search relies solely on the model's internal knowledge. By incorporating retrieved passages, SEARCH-R1 allows the LLM to iteratively refine its reasoning, leading to more informed and accurate responses.

Self-Verification through Iterative Retrieval: We observe that after the second retrieval round, the LLM has already gathered sufficient information to answer the question. However, SEARCH-R1 performs an additional retrieval step to self-verify its conclusion, further reinforcing its confidence in the final response. This phenomenon aligns with findings from LLM reasoning RL without retrieval (Guo et al., 2025), highlighting how reinforcement learning can encourage verification-driven reasoning even in search-augmented settings.

6 Conclusion

In this work, we introduced SEARCH-R1, a novel reinforcement learning framework that enables large language models (LLMs) to interleave self-reasoning with real-time search engine interactions. Unlike existing retrieval-augmented generation (RAG) approaches, which lack flexibility for multi-turn retrieval, or tool-use methods that require large-scale supervised training data, SEARCH-R1 optimizes LLM rollouts through reinforcement learning, allowing autonomous query generation and strategic utilization of retrieved information. Through extensive experiments on seven datasets, we demonstrated that SEARCH-R1 significantly enhances LLMs' ability to tackle complex reasoning tasks requiring real-time external knowledge. Our analysis also provides key insights into RL training strategies for search-augmented reasoning. Looking ahead, future work can explore expanding SEARCH-R1 to support broader search strategies, including more sophisticated reward mechanisms, dynamic retrieval adjustments based on uncertainty, and integration with diverse information sources beyond web search. It is also promising to investigate its applicability to multimodal reasoning tasks.

Acknowledgments

This research was supported in part by Apple PhD Fellowship, in part by US DARPA INCAS Program No. HR0011-21-C0165 and BRIES Program No. HR0011-24-3-0325, in part by the Office of Naval Research contract number N000142412612, in part by NSF grant numbers IIS-19-56151 and 2402873, in part by the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897 and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329, in part by Cisco, and in part by the Center for Intelligent Information Retrieval. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of the sponsors or the U.S. Government.

References

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

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.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint arXiv:2501.11651*, 2025.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

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.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

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)*, pp. 6769–6781, 2020.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 10, 2023.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

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.

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.

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

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

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*, pp. 374–382, 2023.

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.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 7, 2022.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

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.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

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.

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.

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

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.

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.

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

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

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.

Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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*, 2022a.

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

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.

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.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*, 2022.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

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.

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

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason, 2025. Notion Blog.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4): 1–60, 2024.