Tool-Star: Empowering Multi-Tool Collaborative Web Agent via Reinforcement Learning

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

Xiaoxi Li
Jiajie Jin
Renmin University of China
Beijing, China

Hongjin Qian
Beijing Academy of Artificial Intelligence
Beijing, China

Yutao Zhu
Renmin University of China
Beijing, China

Hangyu Mao
Institute of Microelectronics, Chinese Academy of Sciences
Beijing, China

Guorui Zhou
Kuaishou Technology
Beijing, China

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

Ji-Rong Wen
Renmin University of China
Beijing, China

Abstract

Recently, Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities through reinforcement learning (RL). However, enabling LLM-based agents to effectively orchestrate multiple tools remains an open challenge. In this paper, we introduce Tool-Star, an end-to-end agentic post-training framework that empowers LLM-based web agents to strategically interact with external multi-tool environments. Tool-Star begins with a general tool-integrated data synthesis pipeline that combines two complementary sampling strategies to generate tool-use trajectories, followed by quality normalization and difficulty-aware curriculum construction to filter noisy samples and organize training data from easy to hard. We then introduce a two-stage training paradigm for multi-tool collaborative reasoning: (1) cold-start supervised fine-tuning with tool feedback to bootstrap long-horizon tool-augmented reasoning, and (2) a multi-tool self-critic RL algorithm with hierarchical rewards to reinforce effective tool coordination. Experiments across 13 benchmarks demonstrate Tool-Star's effectiveness. Further analyses provide practical insights for optimizing strategic tool use in web agents. The code is available at https://github.com/RUC-NLPIR/Tool-Star.

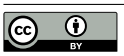
CCS Concepts

• **Information systems** → **Web applications**; • **Computing methodologies** → **Natural language generation**; **Reinforcement learning**.

Keywords

Tool Learning, Agentic Search, Tool-integrated Reasoning

*Corresponding author.



SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809712

ACM Reference Format:
Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2026. Tool-Star: Empowering Multi-Tool Collaborative Web Agent via Reinforcement Learning. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3805712.3809712

1 Introduction

Recent advances in Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, largely driven by large-scale reinforcement learning (RL) [17, 57, 59, 78]. Advanced models such as DeepSeek-R1 [14] and OpenAI-o1 [39] exhibit diverse emergent behaviors in Chain-of-Thought (CoT) reasoning, including deep thinking and self-reflection, thereby improving performance on complex reasoning tasks [38, 46]. However, while CoT reasoning enhances performance on diverse tasks, real-world information-seeking scenarios pose distinct challenges due to the explosive growth of web information and the static knowledge nature of LLMs. These scenarios require LLMs to interact in real-time with multiple tool environments, such as retrieving relevant information via search engines, navigating web pages, and performing precise computations. To address these needs, a series of LLM-based web agents have emerged [32, 33]. These agents strategically conduct web searches during tool-integrated reasoning (TIR) process, thereby achieving in-depth web information seeking [23, 30].

Current foundational work on web agent training primarily focuses on distilling tool-use trajectories from strong models and applying supervised fine-tuning (SFT) to guide weaker models via imitation learning [13, 22, 62, 76]. As large reasoning models exhibit emergent capabilities, subsequent efforts aim to enhance long CoT reasoning through tool-use prompting [3, 27, 32, 70]. However, these approaches rarely enable LLMs to autonomously discover effective tool-use strategies. To address this limitation, recent studies introduce outcome-based rewards [4, 49], extending reinforcement learning to web agent training [2, 11, 23, 34, 43, 52, 65]. While RL-based methods encourage exploration of efficient tool-use behaviors, they often focus on single-tool interaction. In contrast,

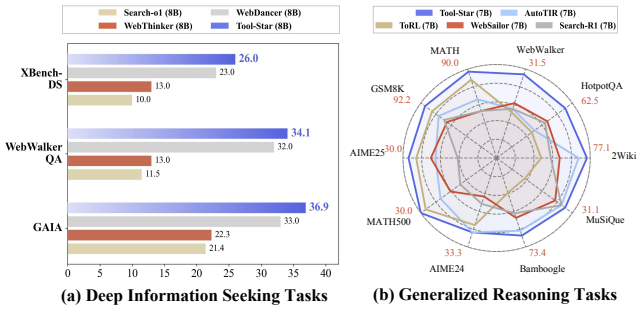


Figure 1: Overview performance of Tool-Star on (a) deep information seeking tasks and (b) generalized reasoning tasks.

real-world web agent tasks frequently require both dynamic information retrieval and accurate computation. This necessitates deeper integration of feedback from multiple tools, especially search engines, web browsers, and code generators throughout the reasoning process. The lack of systematic investigation into multi-tool collaborative reasoning limits reliability and cost control in deployed web agents. In this work, we aim to bridge this gap by addressing the following research questions:

(1) **Rationality and efficiency of tool usage.** How can we enable LLMs to efficiently leverage tools in a reasonable manner for web information seeking?
(2) **Multi-tool collaborative reasoning.** How can we enable LLMs to strategically collaborate with multiple tools during web information seeking?

In this paper, we propose **Tool-Star**, an end-to-end agentic post-training framework that empowers LLM-based web agents to strategically interact with external multi-tool environments. In detail, Tool-Star is designed to foster multi-tool collaborative reasoning by supporting scalable tool integration (including three tools for training and three tools for inference-time scaling). Furthermore, it incorporates systematic innovations in agentic data synthesis and a two-stage training algorithm to improve reasoning efficiency.

To address the scarcity of tool-use data, we first design a scalable **Tool-Integrated Reasoning Data Synthesis Pipeline** that combines tool-integrated prompting with hint-based sampling to automatically generate large-scale tool-use trajectories. We then introduce a quality normalization and difficulty-aware classification process to effectively filter out unreasonable tool-use samples and partition the data in a curriculum-like manner from easy to hard [7, 54]. Leveraging this pipeline, we construct high-quality datasets for both cold-start fine-tuning and RL in a staged manner, laying a solid foundation for subsequent TIR training.

To incentivize the model's capability for multi-tool collaboration, we propose a two-stage agentic training framework that progressively aligns reasoning abilities in an easy-to-hard manner. (1) In the first stage, we introduce a **Cold-Start Supervised Fine-Tuning** strategy, allowing LLMs to initially explore reasoning patterns with feedback from tool invocation. (2) In the second stage, we develop a **Multi-Tool Self-Critic Reinforcement Learning Algorithm**. Unlike prior RL approaches that focus on single-tool usage, our algorithm employs a hierarchical reward mechanism that not only evaluates answer correctness and tool-use format but also assigns rewards for effective multi-tool collaboration. To further improve the model's understanding of this complex reward structure, we interleave a self-critic reward fine-tuning within the standard RL process, facilitating the internalization of reward principles.

To comprehensively evaluate Tool-Star, we conduct experiments on 13 benchmarks spanning deep information-seeking, knowledge-intensive QA and computational reasoning. As shown in Figure 1, Tool-Star consistently outperforms a series of advanced web agents under fair settings. Remarkably, further analyses confirm that Tool-Star achieves strong overall performance while maintaining tool-use efficiency and promoting multi-tool collaborative gains (§4.5), providing a promising solution for training real-world web agents.

In summary, our main contributions are as follows:

- We propose **Tool-Star**, an end-to-end agentic post-training framework that empowers LLM-based web agents to strategically interact with multi-tool environments, thereby fostering multi-tool collaborative reasoning.
- We propose a two-stage agentic training framework that progressively aligns multi-tool collaborative reasoning for LLMs in a curriculum manner: **(1) Cold-start fine-tuning** enables LLMs to explore long-horizon reasoning patterns with tool-use feedback; **(2) Multi-tool self-critic RL** with hierarchical rewards reinforces the LLM's understanding of reward principles and promotes effective multi-tool collaboration.
- To address the scarcity of tool-use data, we propose a scalable data synthesis pipeline that combines two streams of TIR sampling strategies to automatically and scalably generate tool-use trajectories. We further introduce a quality normalization and difficulty-aware classification process to filter out unreasonable samples and organize the dataset from easy to hard.
- Experiments on more than 13 challenging benchmarks consistently demonstrate Tool-Star's superiority over mainstream web agents. Further analyses on tool-use efficiency and collaboration gains offer practical insights for training multi-tool collaborative web agents.

2 Related Work

2.1 Reinforcement Learning for Web Agent.

The emergence of agent reinforcement learning (RL) [80] has enabled the development of general-purpose web agents. Initial efforts [2, 11, 23, 34, 52] established foundations by enabling models to autonomously interact with search engines. Subsequent innovations [6, 19, 53, 64, 65, 72, 83] enhance efficiency and stability of web agent through redesigned reward functions. Recent research [12, 21, 31, 73] explores asynchronous training frameworks for web agents. Tongyi Deep Research [10, 45, 55, 58, 71, 79] broadens the scope by leveraging graph-based data synthesis pipeline and training optimization, thereby broadening the scope of web agent training. To minimize resource consumption, another line of research simulates search engines using large models' generative capabilities for self-alignment [9, 56]. However, empowering multi-tool collaborative reasoning via end-to-end agentic post-training remains largely unexplored. In this paper, we introduce Tool-Star, which empowers LLMs to strategically interact with multi-tool environments and foster collaborative reasoning.

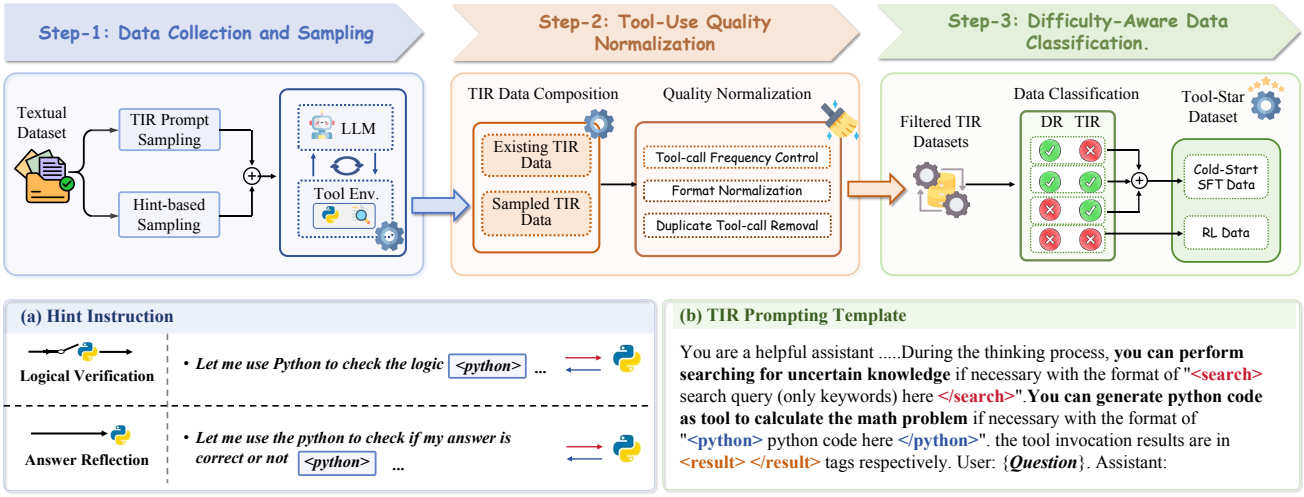


Figure 2: The overview of scalable tool-integrated reasoning data synthesis pipeline.

2.2 Tool-Integrated Reasoning.

Tool-Integrated Reasoning (TIR) enhances LLM reasoning by enabling autonomous invocation of external tools. Existing TIR approaches can be categorized into three streams: **(1) Prompting-based methods** guide models to use tools via carefully crafted prompts without additional training [3, 27, 32, 36, 62, 70]. While easy to implement, they often suffer from instability and limited accuracy in tool usage. **(2) SFT-based methods** apply supervised fine-tuning to teach weaker models tool use by distilling trajectories from stronger models [13, 26, 28, 76]. Though effective, their performance is constrained by the quality of demonstrations and limited generalization beyond seen examples. **(3) RL-based methods** optimize tool use via outcome-driven rewards [2, 11, 23, 33, 34, 51, 52, 56], enabling autonomous discovery of effective strategies. Despite these advances, most work focuses on single-tool settings. Multi-tool collaborative reasoning—requiring coordination across heterogeneous tools—remains underexplored. While OTC [65] and ToolRL [42] have made initial progress, there is still a great gap in developing specialized RL algorithms for multi-tool collaboration.

3 Methodology

Method Overview. Tool-Star is an end-to-end agentic post-training framework that empowers LLMs to autonomously interact with multi-tool environments. As shown in Figure 2 and Figure 3, Tool-Star includes two key components: **(1)** A tool-integrated data synthesis pipeline that generates high-quality trajectories with reasonable tool usage (§3.1); **(2)** A two-stage agentic training paradigm that facilitates multi-tool collaborative reasoning, comprising cold-start fine-tuning and a multi-tool self-critic RL algorithm (§3.2). Below, we will dive into the specifics of our approach.

Problem Formulation. *Multi-tool Collaborative Reasoning* aims to enable LLMs to perform multi-step reasoning through interaction and collaboration with multiple external tools. Specifically, given a task query q and an external tool set $\mathcal{T}$, Tool-Star autonomously invokes tools during the generation of the reasoning chain $\mathcal{R}^c$, guided by the tool-integrated instruction $I^{\mathcal{T}}$. The real-time tool-call feedback $F^{\mathcal{T}}$ is dynamically concatenated into the reasoning chain to facilitate ongoing inference until the final output y is produced. This process can be modeled as:

$$P_\theta(\mathcal{R}^c, y \mid I_{\mathcal{T}}, q, \mathcal{T}) = \underbrace{\prod_{t=1}^{T_c} P_\theta(\mathcal{R}_t^c \mid \mathcal{R}_{<t}^c, I^{\mathcal{T}}, q, \{F^{\mathcal{T}}\}_{<t})}_{\text{Multi-Tool Integrated Reasoning}} \cdot \underbrace{\prod_{t=1}^{T_y} P_\theta(y_t \mid y_{<t}, \mathcal{R}_c, I^{\mathcal{T}}, q)}_{\text{Answer Generation}}, \tag{1}$$

where T_c denotes the token numbers in the reasoning chain $\mathcal{R}^c$, $\mathcal{R}_t^c$ is the token at position t, and $\mathcal{R}_{<t}^c$ represents all tokens generated before t. $\{F^{\mathcal{T}}\}_{<t}$ denotes the feedback of all tool calls prior to t. T_y is the length of the Answer y, with y_t as the token at position t.

Tool Design. In this work, we design six tools to foster multi-tool collaborative reasoning of Tool-Star. During the two-stage training process, we introduce three core tools that enable LLMs to autonomously invoke external functionalities during reasoning:

- **Search Engine:** Executes search queries to retrieve relevant information, supporting both local and web-based search modes.
- **Web Browser:** Parses web search results by visiting URLs, extracting relevant content and summarizing key information.
- **Code Interpreter:** Executes code snippets generated by the LLM in a sandbox environment, returning either the execution results or error messages based on code correctness.

During inference, we introduce three Inference-time scaling tools to optimize the reliability of TIR process:

- **Code Debugger:** Automatically corrects LLM-generated erroneous code by leveraging the original code and compiler error messages to guide revisions.
- **Tool-Use Backtracer:** Locates and rolls back to the reasoning step preceding a failed tool invocation, enabling the model to resume and revise its reasoning path.

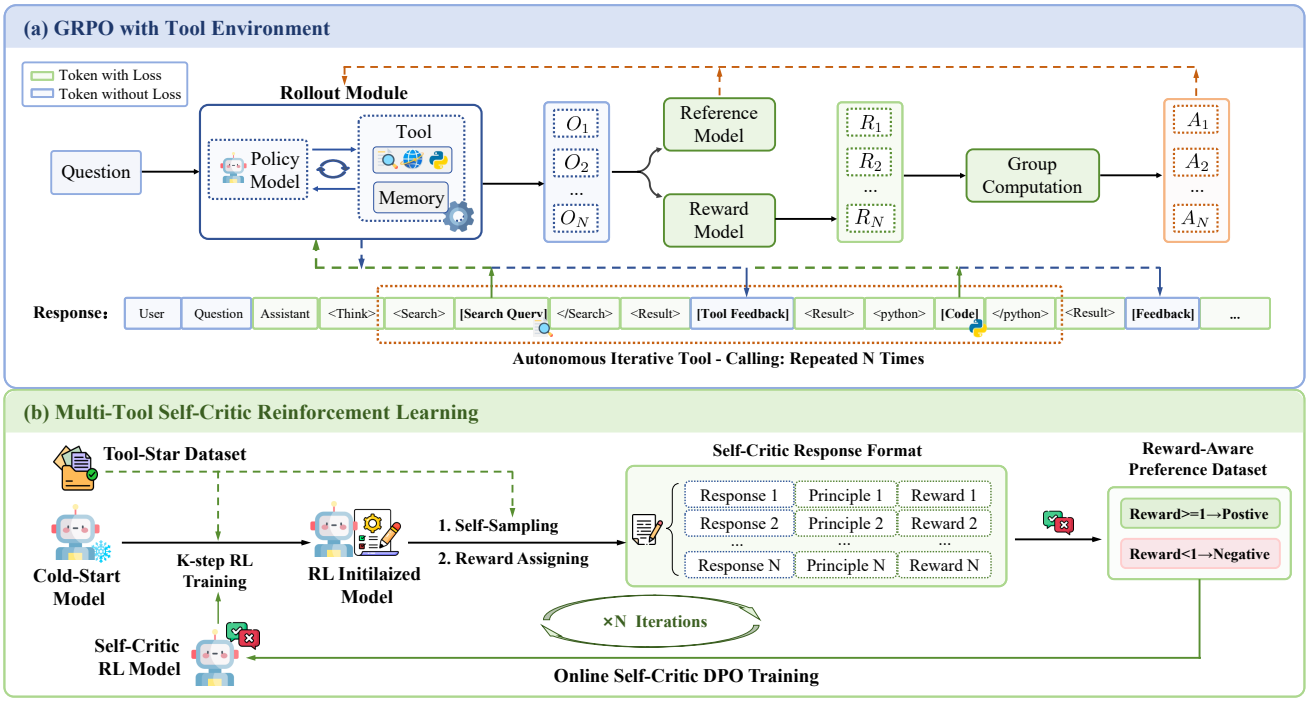


Figure 3: The overview of multi-tool self-critic reinforcement learning algorithm.

- **Reasoning Chain Refiner:** When the output exceeds the maximum length, refiner prunes and optimizes redundant steps in the reasoning process, replacing original reasoning chain with a more concise and coherent version to complete the task.

3.1 Tool-Integrated Reasoning Data Synthesis

In this section, we present our tool-integrated reasoning data synthesis pipeline, aiming to enable automated and scalable construction of high-quality tool-use datasets (Figure 2).

3.1.1 Step-1: Data Collection and Sampling. To balance data scale and diversity while ensuring accessibility, we curate a high-quality training set from open-source knowledge-based and computational reasoning datasets, comprising approximately 90K text-based reasoning data (D_{text}) and 1K existing TIR datasets (D_{tool}). To further expand TIR trajectories, we introduce two complementary sampling strategies:

(1) TIR Prompting-based Sampling. We design a tool-integrated prompt I^T to guide the LLM P_θ in sampling responses for queries in D_{text}. Following the Eq.(1), LLM decodes tool-call requests within special tokens (e.g., <search>, <python>) during trajectory generation. We then automatically parse and extract these requests, invoke external tools to obtain feedback F, and insert the tool feedback—enclosed within <result> and </result> tags—back into the reasoning chain as additional context for subsequent generation steps. This process iterates until either (1) the maximum number of tool calls or length is reached, or (2) the model generates a final answer, enclosed by designated tokens <answer> and </answer>. After inference on the entire D_{text}, we filter for correct samples to obtain the dataset D^P_{tool}.

(2) Hint-based Sampling. To further diversify tool invocation patterns, we employ a hint-based method [29] that inserts hint tool-call tokens into language-only reasoning trajectories. We first prompt the LLM to perform language-only reasoning on queries from D_{text}. Following the START [28], we propose two hint instructions—*Logical Verification* and *Answer Reflection*—to insert tool-invoking hints into the original reasoning chains. As shown in Figure 2, logical verification hints randomly replace uncertain expressions (e.g., *maybe*, *wait*, *not sure*) in the chain, while reflection hints are inserted after the answer. These diverse hints facilitate the model to invoke tools when information is insufficient or after answer generation, enabling information completion and answer verification. After inserting hints, we truncate the original reasoning chain at the hint position, prompting the model to perform TIR reasoning in response to the hint, obtaining the hint-based TIR dataset D^H_{tool}. Finally, we merge two datasets to obtain the final dataset $D^{v1}_{\text{tool}} = \{D^H_{\text{tool}} \cup D^P_{\text{tool}} \cup D_{\text{tool}}\}$.

3.1.2 Step-2: Tool-Use Quality Normalization. To ensure the rationality of tool usage within each sample, we implement the following 3 TIR normalization strategies for tool-use data quality control: **(1) Tool-call Frequency Control:** Remove samples with tool-call frequency exceeding a predefined threshold β to alleviate excessive tool invocation. **(2) Duplicate Tool-call Removal:** Eliminate samples containing redundant tool calls, such as repeated generation of search queries or code snippets in the same response. **(3) Format Normalization:** Standardize tool call formats in reasoning chains by unifying special tokens for invocation, feedback, and final answers, while ensuring balanced usage of start and end tokens. By applying these criteria, we obtain a quality-filtered dataset $D^{\text{v2}}_{\text{tool}}$.

3.1.3 Step-3: Difficulty-Aware Data Classification. Considering the computational overhead of tool-use and the multi-stage nature of TIR training, we argue that a high-quality tool-use dataset should meet the following criteria: **(1) Invoke tools only when necessary:** Tool calls should be avoided when the model is capable of solving the problem through direct reasoning. **(2) Organize samples from easy to hard:** As emphasized in prior ToolRL learning [81], stage-wise training based on sample difficulty is crucial for effective learning.

To achieve the above objectives, we first perform a language-only reasoning pass on each question in $D^{\text{v2}}_{\text{tool}}$, producing direct reasoning results denoted as $D^{\text{v2}}_{\text{text}}$. Based on the correctness of both direct reasoning (DR) and tool-integrated reasoning (TIR), each sample is assigned to one of four categories (Figure 2). For categories 1 and 2, where direct reasoning already yields correct answers, tool use is unnecessary; these samples form the subset $D^{\text{sub}}_{\text{text}}$. For category 3, which highlights clear benefits from tool use, we directly sample from $D^{\text{v2}}_{\text{tool}}$ to construct $D^{\text{sub}}_{\text{tool}}$.

To support a curriculum learning paradigm [7] from easy to hard, we construct a cold-start fine-tuning dataset $D^{\text{SFT}}_{\text{tool}}$ by combining $D^{\text{sub}}_{\text{text}}$ and $D^{\text{sub}}_{\text{tool}}$. For Category-4 samples, which are challenging for both DR and TIR, are treated as hard examples and reserved for reinforcement learning, forming the dataset $D^{\text{RL}}_{\text{tool}}$. This design enables the LLM to acquire basic tool-use capabilities through cold-start fine-tuning, and subsequently generalize to more complex scenarios during the RL phase (e.g. multi-tool collaboration), thereby facilitating a progressive learning trajectory.

3.2 Multi-tool Collaborative Training Stage

In this section, we propose a multi-tool collaborative training stage that generalize the LLM's agentic reasoning capability from a single to multiple tool-use paradigm, and from easy to hard learning.

3.2.1 Cold-Start Supervised Fine-tuning. To equip the LLM with an initial understanding of tool usage for problem solving, given $(x_i, y_i) \in D^{\text{SFT}}_{\text{tool}}$, we apply the standard Supervised Fine-tuning objective on the backbone model P_θ with parameters θ: $\mathcal{L}(\boldsymbol{\theta}) = -\sum_{(x_i,y_i)\in D^{\text{SFT}}_{\text{tool}}} \log P_\theta(y_i \mid x_i)$, where x_i denotes the i-th input. Ultimately, we obtain a cold-start LLM $\hat{\pi}_\theta$ with initial TIR capability.

3.2.2 Multi-Tool Self-Critic Reinforcement Learning. We will introduce how the LLM learn to autonomously invoke tools, including a Code interpreter, a search engine and a web browser agent.

(1) Memory-based Roll-Out with Tools. As shown in Figure 3, we employ multi-tool invocation instructions to guide the model in decoding tool-use requests into special tokens (e.g., <python>) during the roll-out process. Upon detecting these tokens, the corresponding tool is automatically invoked, and the resulting feedback is integrated back into the reasoning chain. To reduce latency caused by frequent tool calls, we incorporate a memory mechanism that caches the mapping between each tool request and its output. This allows the model to retrieve responses for repeated requests directly from memory, thereby improving efficiency.

(2) Hierarchical Reward Design. Reward signals serve as the optimization objective and directly guide the behavior of the policy model during training. Distinct from previous tool-use RL approaches, we not only design correctness and format rewards for LLMs, but also introduce multi-tool collaborative rewards. This design aims to encourage multiple tool usage while maintaining LLM correctness. Specifically, when both the answer and tool invocation format are correct, and the model employs multiple tools (i.e., both <search> and <python> appear in the reasoning chain), an extra reward r_M is granted. Formally, the overall reward R is defined as:

$$R = \begin{cases} \max(\text{Acc.} + r_{\text{M}},\ \text{Acc.}) & \text{if Format is Good \& Acc. } > 0; \\ 0 & \text{if Format is Good \& Acc. } = 0; \\ -1 & \text{otherwise,} \end{cases} \quad (2)$$

where r_M is defined as follows.

$$r_{\text{M}} = \begin{cases} 0.1 & \text{if } \exists(\text{<search>} \ \&\ \text{<python>}); \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

(3) Self-Critic RL Algorithm. Multi-tool RL algorithms involve complex reward structures, making it challenging for LLMs to discover optimal behavior. To address this, we propose a self-critic RL algorithm that enables LLMs to better align with the design principles of reward mechanisms. As shown in Figure 3, **self-critic RL interleaves DPO and GRPO training to facilitate dynamic preference optimization.** This design leverages the strengths of both methods: GRPO treats answers as soft targets to promote multi-tool collaboration, whereas DPO aids in internalizing reward signals by contrasting hard negatives, thereby guiding the model to better capture the underlying reward structure.

As an initial step, we begin by performing K steps of vanilla RL training on the cold-start model $\hat{\pi}_\theta$. To optimize the policy, we adopt Group Relative Policy Optimization (GRPO) [49] as our RL

Algorithm 1 Multi-Tool Self-Critic Reinforcement Learning

Require: Reasoning model π_θ, external tools T, reward function $\mathcal{R}_\phi$, number of training cycles C, GRPO steps per cycle $\mathcal{S}$

Input: Dataset D, task instruction I
Initialize reference model: $\pi^{\text{old}}_\theta \leftarrow \pi_\theta$
for $i \leftarrow \{1, ..., C\}$ **do**
  for step $\leftarrow \{1, ..., \mathcal{S}\}$ **do** ▷ Vanilla GRPO Phase
    Sample mini-batch $D_b \subset D$
    for each query $q \in D_b$ **do**
      Concatenate instruction: $q \leftarrow I \oplus q$
      Sample G TIR trajectories: $\{o_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid q, T)$
      Compute GRPO objective J_θ (Eq. 4)
      Update model parameters: $\theta \leftarrow \theta + \eta \cdot \nabla_\theta J_\theta$
    end for
  end for
  Sample subset $D_{\text{sample}} \subset D$ ▷ Self-Critique DPO Phase
  for each query $q \in D_{\text{sample}}$ **do**
    Concatenate instruction: $q \leftarrow I \oplus q$
    Sample G outputs with tools: $\{o_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid q, T)$
    Identify o_{chosen} (preferred) and o_{reject} (less preferred)
    Compute DPO objective J_θ (Eq. 5)
    Update model parameters: $\theta \leftarrow \theta + \eta \cdot \nabla_\theta J_\theta$
  end for
end for
Output: Fine-tuned model π_θ

algorithm, which estimates the baseline using a group of rollouts.

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{[q\sim D_{\text{tool}}^{\text{RL}},\{o_i\}_{i=1}^{G}\sim\hat{\pi}_{\theta_{\text{old}}}(O|q)]}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left(\frac{\hat{\pi}_\theta(o_{i,t}|q,o_{i,<t})}{\hat{\pi}_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},\ \text{clip}\left(\frac{\hat{\pi}_\theta(o_{i,t}|q,o_{i,<t})}{\hat{\pi}_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})},\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}_{i,t}\right)\right], \tag{4}$$

where ϵ and β are hyper-parameters, q and o denote query and model's output with tools. A_i is the normalized advantage of the i-th rollout in group. Then we obtain an RL-initialized model π_θ^{RL}.

Subsequently, we detail our self-critic reward fine-tuning phrase, which help LLMs better internalize the reward structure. As shown in Figure 3, we start by randomly rejection sampling k examples from the RL training set D_{tool}^{RL} to construct $D_{\text{tool}}^{\text{S}}$ [61, 77]. For each query $q \in D_{\text{tool}}^{\text{S}}$, the RL initialized model π_θ^{RL} self-samples n candidate responses, forming a diverse QA set $D \sim (q, \{a_i\}_{i=1}^{N})$. Notably, our hierarchical rule-based reward function acts as an executable program that automatically assigns a reward label to each response. This enables on-policy reasoning by providing each a_i with a corresponding reward. We then construct a reward-aware dataset $D_{\text{tool}}^{\text{critic}} = \{(x_i, y_i)\}_{i=1}^{N}$, where each input x_i is a query q, and each output y_i includes a candidate response a_i, its reasoning trace p_i, and reward score r_i. We treat samples with $r_i \geq 1$ as positive and those with $r_i < 1$ as negative, forming a preference dataset $(x, y_w, y_l) \sim D_{\text{tool}}^{\text{critic}}$. Finally, we fine-tune the RL-initialized model using DPO objective as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta^{\text{RL}};\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta^{\text{RL}}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta^{\text{RL}}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \tag{5}$$

where the reference model π_{ref} is set to π_θ^{RL} initially and remains fixed throughout training. β is a hyperparameter and σ is the sigmoid function. $\mathcal{L}_{\text{DPO}}$ encourages higher log-probability for preferred responses y_w over dispreferred ones y_l.

Therefore, we interleave self-critic reward fine-tuning every k steps during standard RL training. This iterative process allows the model to progressively learn behavioral distinctions via self-sampling and evaluation, implicitly capturing the hierarchical reward structure. The full algorithm is detailed in Algorithm 1.

3.3 Multi-tool Collaborative Inference

During inference, we follow Eq.(1), enabling the model to employ the *search engine*, *web browser agent* and *Python interpreter* autonomously. To improve TIR robustness, we introduce three inference-time mechanisms targeting common failure scenarios:

- **Code Execution Error:** When generated code contains syntax errors, the *Code Debugger* uses compiler messages and the original code to produce executable fixes.
- **Tool Invocation Failure:** If tool calls yield no useful output or debugging fails, the *Backtracer* identifies the statement before the tool-call token and restarts inference from there.
- **Inference Length Overflow:** When tool outputs exceed the context limit, the *Chain Refiner* compresses and restructures the reasoning chain by removing redundant or irrelevant content, allowing continued inference without tools.

These mechanisms enhance tool invocation reliability and provide an effective inference-time scaling strategy.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate the tool-use capabilities of our model, we conduct experiments on two types of reasoning datasets: (1) **Knowledge reasoning benchmarks**, including four Wikipedia-based open-domain QA tasks: HotpotQA [74], 2WikiMultihopQA [16], Musique [63] and Bamboogle [41]. (2) **Computational reasoning benchmarks**, including AIME2024, AIME2025 [1], MATH500 [35], MATH [15], and GSM8K. **(3) Deep information seeking tasks**, including General AI Assistant (GAIA) [37], WebWalkerQA (WebWalker)[69], Humanity's Last Exam (HLE) [40], and xbench DeepSearch split [1]. For all tasks, we follow previous work [32] and extract answers from the output enclosed in `\box{}`.

Baselines. To effectively evaluate the efficacy of Tool-Star, we compare the following 3 categories of baselines: **(1) Proprietary Models**: Qwen2.5 [47], Llama3.2 [8], QwQ [60], DeepSeek-R1 [14], GPT-4o [20]and o1-preview [20]. **(2) Code-integrated Models:** ToRL [34], DotaMath [26], ReTool [11], START [28]. **(3) Advanced Search Agents:** Vanilla RAG [25], ReAct [75], Search-o1 [32], Search-R1 [23], WebSailor [30] and WebDancer [68] and Webthinker [33]. **(4) Multi-tool Enhanced Baselines**, including prompt-based approaches, ReCall, IKEA, SMART, AutoTIR, [2, 18, 44, 67].

Evaluation Metrics. For deep-information seeking tasks, we follow Search-o1 [32]'s evaluation setting and employ strong LLM-based judging (Qwen3-235B-A22B) to ensure answer correctness. We randomly sampled 200 validation samples and confirmed that the agreement rate between LLM judge and human evaluation is 100%. For open-domain QA and reasoning tasks, we adopt token-level F1 score and accuracy as the evaluation metric. To assess tool usage efficiency, we propose the *Tool-use Efficiency* (T_E) metric: $T_E = \frac{1}{N}\sum_{i=1}^{N}\frac{S_i}{T_i^c}$, where N denotes the total number of evaluation datasets, S_i represents the number of correctly answered samples using tools on the i-th dataset, and T_i^c is the total number of tool-invoked samples in that dataset. A higher T_E value indicates more effective tool utilization in producing correct answers.

Implementations. Our training is conducted using the LLaMA-Factory [82] and VERL frameworks [50]. For SFT stage, we adopt DeepSpeed ZeRO-3 [48] and FlashAttention2 [5] to accelerate training. The model is trained for 3 epochs with a batch size of 128, weight decay of 0.1, and mixed-precision (BF16) training. The RL stage runs for 2 epochs on 8 NVIDIA A800 GPUs, with 10k examples randomly sampled from D_{tool}^{RL}. For challenging reasoning tasks, we employ E5-base-v2 [66] to retrieve the top 10 documents for each query from the 2018 Wikipedia dump [24]. For deep information seeking tasks, we use the Bing search engine to retrieve 10 web pages, browse each page contents from the URLs, and employ a browser agent of the same size as the reasoning model to summarize

[1] https://huggingface.co/datasets/AI-MO/aimo-validation-aime

Table 1: Overall results on 10 challenging QA and reasoning benmarks. For methods sharing the same backbone, the top two results are highlighted in bold and underlined. and represent methods with code interpreter and search tool, respectively. For fairness, we reproduce WebSailor-7B using Wikipedia-based data and report the RAG setting (Top-5 documents) for code-assistant models in knowledge-intensive QA tasks. The dataset abbreviation: 2Wiki. refers to 2wikiMultiHopQA.

Method	Size	Knowledge-Intensive QA					Computational Reasoning					Avg.
		WebWalker	HotpotQA	2Wiki.	MuSiQue	Bamboogle	AIME24	AIME25	MATH500	GSM8K	MATH	
Qwen2.5-Instruct	7B	0.5	26.1	25.6	7.9	36.5	10.0	10.0	70.6	90.2	82.0	35.9
Llama3.1-Instruct	8B	4.0	16.2	13.7	7.4	23.2	3.3	3.3	43.3	78.4	60.6	25.3
Single-Tool Integrated Reasoning Methods												
RAG	7B	9.1	35.1	25.0	7.5	23.2	13.3	3.3	70.8	86.8	80.0	35.4
Search-o1	7B	9.5	41.1	35.4	13.2	39.8	6.7	10.0	61.8	80.2	73.6	37.1
Search-R1	7B	12.5	46.6	40.0	22.8	47.4	16.7	6.7	63.8	82.4	81.2	42.1
WebThinker	7B	14.1	47.5	42.8	22.3	48.0	16.7	10.0	66.0	84.8	82.5	43.5
WebSailor	7B	15.2	47.5	44.3	20.5	49.7	13.3	13.3	74.4	80.8	81.2	44.0
ToRL	7B	10.9	41.3	35.4	9.5	36.9	26.6	30.0	80.2	89.2	87.4	44.7
DotaMath	7B	16.5	26.2	21.7	6.5	28.6	16.7	13.3	62.2	86.7	64.8	34.3
ReTool	7B	8.5	31.5	29.0	11.1	35.8	23.3	30.0	78.4	86.2	84.8	41.9
START	7B	8.5	30.2	28.0	10.2	35.8	23.3	26.6	79.6	88.4	83.5	41.4
Multi-Tool Integrated Reasoning Methods												
Multi-Tool Prompting	7B	15.5	21.1	23.8	9.9	25.5	6.7	10.0	68.2	64.6	78.2	32.4
IKEA	7B	11.4	30.5	34.7	14.8	35.8	10.0	10.0	63.6	73.4	80.2	36.4
SMART	7B	9.5	25.4	21.7	11.8	40.7	6.6	6.6	39.4	59.8	70.5	29.2
AutoTIR	7B	12.5	42.4	53.4	22.8	56.3	**30.0**	20.0	72.6	85.5	83.5	47.9
ReCall	7B	15.2	51.9	54.0	**25.0**	55.5	16.7	16.7	54.6	79.8	73.2	44.3
Tool-Star (Qwen2.5)	7B	**29.0**	**54.7**	**57.4**	24.1	**58.8**	**30.0**	**33.3**	**83.2**	**93.2**	**89.0**	**55.1**
Tool-Star (Llama3)	8B	26.5	49.3	50.8	21.2	47.7	16.7	10.0	56.8	82.9	74.6	43.6

Table 2: Analysis of Tool-Star with multi-tool collaboration. , , denote code interpreter, search and browser tool.

Method	GAIA	HLE	Webwalker	MTC
Qwen3-8B	20.4	4.6	2.0	-
Tool-Star( )	25.4	6.6	8.0	0
Tool-Star ( )	32.8	7.0	23.5	0
Tool-Star (+)	35.5	8.6	34.9	39
Tool-Star (+ +)	**39.8**	**10.0**	**38.5**	**92**

Table 3: General Ablation study on Tool-Star(3B).

Method	Knowledge.		Computational.	
	HQA.	Bamb.	GSM8K	MATH
Tool-Star (3B)	51.9	52.5	85.0	82.6
w/o Cold-Start SFT	43.5 (-8.4)	40.8 (-11.7)	77.8 (-7.2)	74.2 (-8.4)
w/o RL stage	47.5 (-4.4)	43.9 (-8.6)	80.2 (-4.8)	78.4 (-4.2)
w/o Self-Critic RL	49.8 (-2.1)	48.3 (-4.2)	82.8 (-2.2)	77.8 (-4.8)
w/o Multi-tool Reward	50.4 (-1.5)	50.3 (-2.2)	83.1 (-1.9)	80.2 (-2.4)

the information. All results are averaged over three independent runs for reproducibility.

4.2 Main Results on General Reasoning Tasks

Our main results are presented in Table 1. Overall, Tool-Star consistently outperforms all baselines across 10 challenging reasoning tasks, achieving average accuracies of 55.1% on Qwen2.5-7B and 43.6% on Llama3-8B, decisively establishing the superiority of our approach. We identify the following key insights:

(1) TIR prompting fails to explore better tool-use behavior. Focusing on Search-o1 and Multi-Tool Prompting, their performance on comprehensive reasoning tasks remains suboptimal, with average accuracies of 37.1% and 32.4%, respectively. Notably, Multi-Tool Prompting exhibits lower consistency compared to its backbone model (32.4% vs. 35.9% for Qwen2.5-Instruct). This reveals that relying solely on prompt engineering to elicit tool usage is insufficient for guiding LLMs toward effective tool utilization.

(2) Single-tool RL-based methods exhibit strong domain specialization but limited generalizability. RL-based search agents (e.g., Search-R1, WebSailor) perform well on knowledge-intensive tasks, achieving up to 47.5% on HotpotQA and 44.3% on 2Wiki. However, their performance on computational reasoning tasks remains limited, with Search-R1 achieving only 6.7% on AIME25. Conversely, code-assistant methods like ToRL achieve strong performance on computational tasks (44.7% average, with 87.4% on MATH) but underperform on knowledge-based tasks. These trends underscore the specialization bias of single-tool RL methods and their limited cross-domain transferability.

(3) Tool-Star demonstrates versatile agentic reasoning with robust plug-and-play capability. Tool-Star(7B) consistently outperforms both single-tool and multi-tool baselines, achieving 55.1%

Table 4: Overall performance on deep information seeking tasks. The best results are indicated in bold, and the second-best results are underlined. Results from larger or closed-source models are presented in gray for reference.

Method	General AI Assistant				WebWalkerQA				Humanity's Last Exam				XBench-DS	FRAMES
	Lv.1	Lv.2	Lv.3	Avg.	Easy	Med.	Hard	Avg.	NS	CE	SF	Avg.	Avg.	Avg.
Direct Reasoning (>=32B)														
Qwen3-32B-thinking	26.2	12.1	0	14.9	6.9	1.1	2.9	3.1	14.6	9.8	8.4	12.6	14.0	26.0
DeepSeek-R1-32B	21.5	13.6	0.0	14.2	7.5	1.4	4.2	3.8	6.6	5.1	6.5	6.4	10.0	23.8
QwQ-32B	30.9	6.5	5.2	18.9	7.5	2.1	4.6	4.3	11.5	7.3	5.2	9.6	10.7	28.8
GPT-4o	23.1	15.4	8.3	17.5	6.7	6.0	4.2	5.5	2.7	1.2	3.2	2.6	18.0	44.6
DeepSeek-R1-671B	40.5	21.2	5.2	25.2	5.0	11.8	11.3	10.0	8.5	8.1	9.3	8.6	32.7	45.6
o1-preview†	-	-	-	-	11.9	10.4	7.9	9.9	12.9	8.1	6.6	11.1	-	-
Deep Information Seeking Methods (Qwen3-8B)														
Vanilla RAG	28.2	15.4	**16.7**	20.4	8.9	10.7	9.9	10.0	5.1	1.6	12.9	5.8	8.0	18.8
Search-o1	35.9	15.4	0.0	21.4	6.7	15.5	9.7	11.5	**7.6**	2.7	5.3	6.4	10.0	19.2
WebThinker	43.6	11.5	0.0	22.3	6.7	13.1	16.9	13.0	7.3	4.0	6.3	6.6	13.0	21.4
ReAct	35.9	17.3	8.3	23.3	8.9	16.7	18.3	15.5	4.2	4.0	6.3	4.6	16.0	21.1
WebDancer	48.7	28.0	8.3	33.0	31.1	**35.7**	**28.2**	32.0	6.4	4.0	10.5	6.8	23.0	44.2
Tool-Star(Ours)	**51.3**	**34.6**	8.3	**36.9**	**43.1**	35.0	27.1	**34.1**	7.3	**6.7**	**15.8**	**8.8**	**26.0**	**47.8**
Deep Information Seeking Methods (Qwen3-14B)														
Vanilla RAG	38.5	19.2	8.3	25.2	17.8	13.1	11.3	13.5	5.5	6.3	9.4	6.0	15.0	31.4
Search-o1	48.7	23.1	0.0	30.1	11.1	21.4	16.9	17.5	6.4	4.0	10.5	6.8	21.0	39.8
WebThinker	48.7	26.9	8.3	33.0	13.3	23.8	18.3	19.5	7.0	4.0	9.5	7.0	23.0	40.8
ReAct	48.7	25.0	8.3	32.0	11.1	20.2	12.7	15.5	5.8	5.3	10.5	6.6	20.0	37.6
WebDancer	48.7	**34.6**	16.6	37.8	**47.7**	33.6	15.6	36.8	7.9	6.7	12.6	8.6	25.0	49.1
Tool-Star(Ours)	**53.9**	**34.6**	**16.7**	**39.8**	35.6	**42.9**	**35.2**	**38.5**	**10.3**	**10.7**	**13.7**	**10.0**	**28.0**	**54.6**

average accuracy across 10 datasets while maintaining competitive performance on individual tasks (e.g., 83.2% on MATH500). Notably, it delivers substantial improvements across different backbones and parameter scales, with average gains of 19.2% on Qwen2.5-7B and 18.3% on Llama3-8B . These results underscore Tool-Star's generality and strong adaptability across both models and tasks.

4.3 Main Results on Deep Information Seeking

To validate the effectiveness of Tool-Star in challenging deep information seeking tasks, we train Qwen3-8B and Qwen3-14B models combined with Tool-Star and compare them with advanced web agents. As shown in Table 4, we derive the following insights:

(1) Limitations of Advanced Large Models: Both closed-source and large-parameter open-source LLMs (e.g., GPT-4o and DeepSeek-R1-671B) perform poorly in challenging deep information seeking scenarios. GPT-4o achieves only 17.5% and 5.5% on General AI Assistant and WebWalkerQA, Notably, most large models achieve less than 10% average accuracy on the Humanity's Last Exam (GPT-4o: 2.6%). This indicates that relying solely on model internal knowledge is insufficient for complex agentic search.

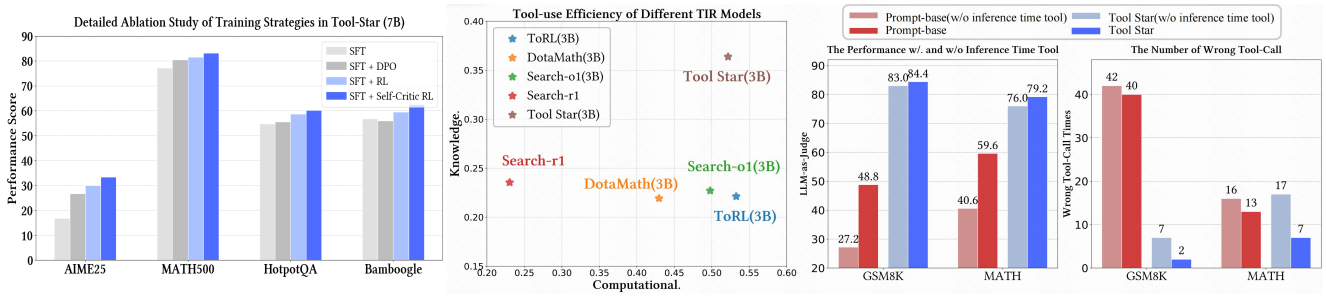
(2) Strong Performance of Tool-Star in Deep Information Seeking: Compared to robust web agents, Tool-Star demonstrates exceptional performance across all deep information seeking benchs. Tool-Star (8B) achieves 36.9%, 34.1%, and 8.8% on GAIA, WebWalkerQA and HLE, respectively, outperforming strong RL-based search agent such as WebDancer and Search-R1. The performance advantage becomes more pronounced as parameter increases, with Tool-Star (14B) showing substantial improvements over its 8B counterpart, particularly on WebWalkerQA where the gain exceeds 4%. This highlights its ability to handle complex agentic search scenarios that require sophisticated multi-tool orchestration.

(3) Superior Generalization Across Diverse Benchmarks: Beyond its outstanding performance on 10 challenging reasoning tasks(Table 1), Tool-Star consistently outperforming baselines across diverse deep information seeking scenarios. Notably, Tool-Star achieves comprehensive performance across deep web information seeking tasks (GAIA, WebWalkerQA), reasoning-search integration tasks (HLE), Chinese search domains (XBench-DR), and factual search scenarios (FRAMES), fully demonstrating its versatility. These results underscore Tool-Star's strong generalization ability across diverse deep information seeking scenarios.

4.4 Ablation Study

General Ablation Study. In this section, we conduct a systematic ablation study in Table 3, where "w/o" denotes removing a specific component. The results demonstrate that: **(1)** Removing any single module leads to performance degradation, highlighting the critical role of each part of design. **(2)** Excluding either results in significant performance drops. the Cold-Start phase helps the model initially acquire tool-use capabilities, while the RL phase promotes generalization in multi-tool reasoning. Both stages are indispensable and complementary. **(3)** Incorporating hierarchical rewards and a self-critic mechanism on vanilla RL consistently brings further improvements, confirming the effectiveness of our tailored RL strategy.

Detailed Ablation on Training Strategies. To thoroughly evaluate the effectiveness of Tool-Star's complete training strategy



(a) (b) (c) (d)

Figure 4: (a) Detailed Ablation on Training Strategies of Tool-Star (7B). (b): Tool-use efficiency comparison across TIR methods. (c): Performance analysis of TIR methods with inference-time tool. (d): Error statistics of tool invocation.

(cold-start SFT+self-critic RL), we conduct a comparative analysis of four training paradigms, as shown in Figure 4 (a). The results indicate that incorporating DPO after SFT yields inferior performance compared to vanilla RL, underscoring the superiority of on-policy RL in modeling tool-use behaviors. Importantly, our self-critic RL, which interleaves DPO and GRPO, achieves consistently strong results across all four datasets, suggesting that it effectively integrates the complementary strengths of RL and DPO for dynamic preference optimization.

4.5 Quantitative Analysis

Multi-Tool Collaboration Gains Analysis. To further verify Tool-Star's self-critic RL mechanism truly brings collaborative gains across multiple tools, we compare the performance of Tool-Star (8B) under different tool combinations setting while maintaining data consistency. Notably, *MTC* denotes the number of multiple tool-call samples across three datasets. We select GAIA, HLE, and WebWalkerQA as evaluation benchmarks, as these challenging datasets are naturally suited for multi-tool application scenarios. As shown in Table 2, when the model simultaneously combines search and code tools, it demonstrates progressive performance improvements compared to single-tool configurations. Furthermore, as the number of tools increasing ($2 \rightarrow 3$), the MTC metric shows a substantial increase ($39 \rightarrow 92$), and Tool-Star achieves notable performance gains across all datasets correspondingly. This result indicates a positive correlation between multi-tool collaboration times and model performance, validating that Tool-Star does not simply apply tools in parallel, but rather learns collaborative multi-tool usage paradigms for achieving collaborative gains.

Tool-Use Efficiency Analysis. To validate whether Tool-Star can efficiently invoke tools during stepwise reasoning, we compare the Tool-use Accuracy T_E of various TIR methods on knowledge-intensive and computation-based reasoning datasets, as shown in Figure 4 (b). Results indicate that while Search-R1 and ToRL achieve high accuracy in knowledge and computation tasks respectively, they underperform in the other domain—highlighting the tendency of single-tool RL methods to specialize in specific reasoning types. In contrast, Tool-Star consistently maintains high tool-use accuracy across both task types and baselines, demonstrating its efficiency in tool utilization.

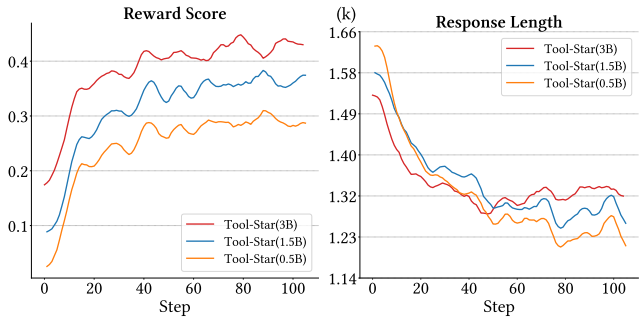


Figure 5: Training curves of reward and response length for models with different parameters.

Inference-Time Tool-Use Analysis. To verify the effectiveness of inference-time tool design, we further analyze its impact on the performance of Tool-Star and DotaMath across two datasets. As shown in Figures 4 (c) and (d), both models exhibit a notable reduction in tool-use errors during reasoning after applying inference-time tool optimizations, accompanied by significant performance improvements. These results not only underscore the detrimental effect of incorrect tool usage on reasoning accuracy, but also reinforces the motivation for incorporating inference-time tools. Notably, the performance gain for DotaMath is substantially larger than that for Tool-Star, with improvements exceeding 20% on GSM8K. This suggests that inference-time tools are particularly beneficial for models with weaker coding capabilities, while models with stronger coding proficiency, such as Tool-Star, naturally make fewer tool-related errors and thus have less room for improvement.

Scaling Analysis on Parameter Scales. To investigate the effectiveness of Tool-Star across different parameter scales, we display the RL training curves of Qwen2.5 with 0.5B, 1.5B, and 3B parameters in Figure 5. Our key observations are as follows: (1) All models achieve good reward scores at the beginning of training (step 0), benefiting from the cold-start SFT stage, which significantly reduces the exploration burden in early RL; (2) As training progresses, the average reward scores steadily improve across all model sizes, indicating that our self-critic RL framework further improve TIR capabilities on top of the cold-start initialization. (3) Notably, the average reward shows signs of emergent improvement around step

10, while no clear inflection point (i.e., "aha moment") is observed in response length. Instead, response lengths gradually stabilize, suggesting convergence toward optimized response patterns.

5 Conclusion

In this paper, we introduce Tool-Star, an end-to-end agentic post-training framework that empowers LLM-based web agents to strategically interact with external multi-tool environments. To address the scarcity of tool-use data, Tool-Star employs a scalable TIR data synthesis pipeline, incorporating normalization and a difficulty-aware data classification process. Furthermore, we propose a two-stage agentic training framework to enhance multi-tool collaborative reasoning in LLMs, consisting of a Cold-Start Fine-tuning phase and a Multi-Tool Self-Critic RL stage. This framework progressively fosters effective multi-tool collaboration and strengthens the LLM's understanding of reward principles. Experiments across 13 challenging benchmarks consistently demonstrate Tool-Star's superiority over mainstream web agents. Further analyses on tool-use efficiency and collaboration gains offer practical insights for training multi-tool collaborative web agents.

Acknowledgments

This work was supported by National Natural Science Foundation of China No. 625B2178 and No. 62272467. This work was supported by the Beijing Major Science and Technology Project under Contract no. Z251100008425002. The work was partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

References

[1] Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. 2025. xbench: Tracking Agents Productivity Scaling with Profession-Aligned Real-World Evaluations. *arXiv preprint arXiv:2506.13651* (2025).

[2] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. arXiv:2503.19470 [cs.AI] https://arxiv.org/abs/2503.19470

[3] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Trans. Mach. Learn. Res.* 2023 (2023). https://openreview.net/forum?id=YfZ4ZPt8zd

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *CoRR* abs/2110.14168 (2021). arXiv:2110.14168 https://arxiv.org/abs/2110.14168

[5] Tri Dao. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *CoRR* (2023).

[6] Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, Yang Qin, Yuan Wang, Quanxing Zha, Sunhao Dai, and Changhua Meng. 2025. Atom-Searcher: Enhancing Agentic Deep Research via Fine-Grained Atomic Thought Reward. *CoRR* abs/2508.12800 (2025). arXiv:2508.12800 doi:10.48550/ARXIV.2508.12800

[7] Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 177–198. doi:10.18653/V1/2024.ACL-LONG.12

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

[9] Yuchen Fan, Kaiyan Zhang, Heng Zhou, Yuxin Zuo, Yanxu Chen, Yu Fu, Xinwei Long, Xuekai Zhu, Che Jiang, Yuchen Zhang, Li Kang, Gang Chen, Cheng Huang, Zhizhou He, Bingning Wang, Lei Bai, Ning Ding, and Bowen Zhou. 2025. SSRL: Self-Search Reinforcement Learning. *CoRR* abs/2508.10874 (2025). arXiv:2508.10874 doi:10.48550/ARXIV.2508.10874

[10] Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. 2025. Towards General Agentic Intelligence via Environment Scaling. *arXiv preprint arXiv:2509.13311* (2025).

[11] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. ReTool: Reinforcement Learning for Strategic Tool Use in LLMs. arXiv:2504.11536 [cs.CL] https://arxiv.org/abs/2504.11536

[12] Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond Ten Turns: Unlocking Long-Horizon Agentic Search with Large-Scale Asynchronous RL. arXiv:2508.07976 [cs.CL] https://arxiv.org/abs/2508.07976

[13] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=Ep0TtjVoap

[14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.). https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html

[16] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 6609–6625. doi:10.18653/V1/2020.COLING-MAIN.580

[17] Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 1049–1065. doi:10.18653/V1/2023.FINDINGS-ACL.67

[18] Ziyang Huang, Xiaowei Yuan, Yiming Ju, Jun Zhao, and Kang Liu. 2025. Reinforced Internal-External Knowledge Synergistic Reasoning for Efficient Adaptive Search Agent. *CoRR* abs/2505.07596 (2025). arXiv:2505.07596 doi:10.48550/ARXIV.2505.07596

[19] Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. 2025. Reinforcement Learning with Rubric Anchors. *CoRR* abs/2508.12790 (2025). arXiv:2508.12790 doi:10.48550/ARXIV.2508.12790

[20] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).

[21] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. 2025. VerlTool: Towards Holistic Agentic Reinforcement Learning with Tool Use. *arXiv preprint arXiv:2509.01055* (2025).

[22] Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2024. RAG-Star: Enhancing Deliberative Reasoning with Retrieval Augmented Verification and Refinement. *CoRR* abs/2412.12881 (2024). arXiv:2412.12881 doi:10.48550/ARXIV.2412.12881

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

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

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and

Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[26] Chengpeng Li, Guanting Dong, Mingfeng Xue, Ru Peng, Xiang Wang, and Dayiheng Liu. 2024. DotaMath: Decomposition of Thought with Code Assistance and Self-correction for Mathematical Reasoning. *CoRR* abs/2407.04078 (2024). arXiv:2407.04078 doi:10.48550/ARXIV.2407.04078

[27] Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. 2024. Chain of Code: Reasoning with a Language Model-Augmented Code Emulator. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=vKtomqlSxm

[28] Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025. START: Self-taught Reasoner with Tools. *CoRR* abs/2503.04625 (2025). arXiv:2503.04625 doi:10.48550/ARXIV.2503.04625

[29] Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025. START: Self-taught Reasoner with Tools. *CoRR* abs/2503.04625 (2025). arXiv:2503.04625 doi:10.48550/ARXIV.2503.04625

[30] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. WebSailor: Navigating Super-human Reasoning for Web Agent. *CoRR* abs/2507.02592 (2025). arXiv:2507.02592 doi:10.48550/ARXIV.2507.02592

[31] Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, He Zhu, Dingfeng Shi, Piaohong Wang, Yeyi Guan, Xiangru Tang, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Jiaheng Liu, Ge Zhang, and Wangchunshu Zhou. 2025. Chain-of-Agents: End-to-End Agent Foundation Models via Multi-Agent Distillation and Agentic RL. *CoRR* abs/2508.13167 (2025). arXiv:2508.13167 doi:10.48550/ARXIV.2508.13167

[32] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. *CoRR* abs/2501.05366 (2025). arXiv:2501.05366 doi:10.48550/ARXIV.2501.05366

[33] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. *arXiv preprint arXiv:2504.21776* (2025).

[34] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. ToRL: Scaling Tool-Integrated RL. *CoRR* abs/2503.23383 (2025). arXiv:2503.23383 doi:10.48550/ARXIV.2503.23383

[35] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=v8L0pN6EOi

[36] Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. 2025. OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning. *CoRR* abs/2502.11271 (2025). arXiv:2502.11271 doi:10.48550/ARXIV.2502.11271

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

[38] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413* (2024).

[39] OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms

[40] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Daron Anderson, Tung Nguyen, Mobeen Mahmood, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Jessica P. Wang, Pawan Kumar, Oleksandr Pokutnyi, Robert Gerbicz, Serguei Popov, John-Clark Levin, Mstyslav Kazakov, Johannes Schmitt, Geoff Galgon, Alvaro Sanchez, Yongki Lee, Will Yeadon, Scott Sauers, Marc Roth, Chidozie Agu, Søren Riis, Fabian Giska, Saiteja Utpala, Zachary Giboney, Gashaw M. Goshu, Joan of Arc Xavier, Sarah-Jane Crowson, Mohinder Maheshbhai Naiya, Noah Burns, Lennart Finke, Zerui Cheng, Hyunwoo Park, Francesco Fournier-Facio, John Wydallis, Mark Nandor, Ankit Singh, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Darling Duclosel, Jungbae Nam, Jennifer Zampese, Ryan G. Hoerr, Aras Bacho, Gautier Abou Loume, Abdallah Galal, Hangrui Cao, Alexis C. Garretson, Damien Sileo, Qiuyu Ren, Doru Cojoc, Pavel Arkhipov, Usman Qazi, Lianghui Li, Sumeet Motwani, Christian Schröder de Witt, Edwin Taylor, Johannes Veith, Eric Singer, Taylor D. Hartman, Paolo Rissone, Jaehyeok Jin, Jack Wei Lun Shi, Chris G. Willcocks, Joshua Robinson, Aleksandar Mikov, Ameya Prabhu, Longke Tang, Xavier Alapont, Justine Leon Uro, Kevin Zhou, Emily de Oliveira Santos, Andrey Pupasov Maksimov, Edward Vendrow, Kengo Zenitani, Julien Guillod, Yuqi Li, Joshua Vendrow, Vladyslav Kuchkin, and Ng Ze-An. 2025. Humanity's Last Exam. *CoRR* abs/2501.14249 (2025). arXiv:2501.14249 doi:10.48550/ARXIV.2501.14249

[41] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 5687–5711. doi:10.18653/V1/2023.FINDINGS-EMNLP.378

[42] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958* (2025).

[43] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. ToolRL: Reward is All Tool Learning Needs. arXiv:2504.13958 [cs.LG] https://arxiv.org/abs/2504.13958

[44] Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. SMART: Self-Aware Agent for Tool Overuse Mitigation. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. 4604–4621. https://aclanthology.org/2025.findings-acl.239/

[45] Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, et al. 2025. WebResearcher: Unleashing unbounded reasoning capability in Long-Horizon Agents. *arXiv preprint arXiv:2509.13309* (2025).

[46] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. 2024. O1 Replication Journey: A Strategic Progress Report–Part 1. *arXiv preprint arXiv:2410.18982* (2024).

[47] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] https://arxiv.org/abs/2412.15115

[48] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters *(KDD '20)*.

[49] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *CoRR* abs/2402.03300 (2024). arXiv:2402.03300 doi:10.48550/ARXIV.2402.03300

[50] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*. 1279–1297. doi:10.1145/3689031.3696075

[51] Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic Reasoning and Tool Integration for LLMs via Reinforcement Learning. *arXiv preprint arXiv:2505.01441* (2025).

[52] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning. *CoRR* abs/2503.05592 (2025). arXiv:2503.05592 doi:10.48550/ARXIV.2503.05592

[53] Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-Searcher++: Incentivizing the Dynamic Knowledge Acquisition of LLMs via Reinforcement Learning. *CoRR* abs/2505.17005 (2025). arXiv:2505.17005 doi:10.48550/ARXIV.2505.17005

[54] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum Learning: A Survey. *Int. J. Comput. Vis.* 130, 6 (2022), 1526–1565. doi:10.1007/S11263-022-01611-X

[55] Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, et al. 2025. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310* (2025).

[56] Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. ZeroSearch: Incentivize the Search Capability of LLMs without Searching. arXiv:2505.04588 [cs.CL] https://arxiv.org/abs/2505.04588

[57] Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, Yue Wu, Wenhai Wang, Junsong Chen, Zhangyue Yin, Xiaozhe Ren, Jie Fu, Junxian He, Wu Yuan, Qi Liu, Xihui Liu, Yu Li, Hao Dong, Yu Cheng, Ming Zhang, Pheng-Ann Heng, Jifeng Dai, Ping Luo, Jingdong Wang, Ji-Rong Wen, Xipeng Qiu, Yike Guo, Hui Xiong, Qun Liu, and Zhenguo Li. 2023. A Survey of Reasoning with Foundation Models. *CoRR* abs/2312.11562 (2023). arXiv:2312.11562 doi:10.48550/ARXIV.2312.11562

[58] Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. WebShaper: Agentically Data Synthesizing via Information-Seeking Formalization. *CoRR* abs/2507.15061 (2025). arXiv:2507.15061 doi:10.48550/ARXIV.2507.15061

[59] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599* (2025).

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

[61] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). arXiv:2302.13971 doi:10.48550/ARXIV.2302.13971

[62] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* (2022).

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

[64] Hongru Wang, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, and Kam-Fai Wong. 2025. Toward a Theory of Agents as Tool-Use Decision-Makers. *CoRR* abs/2506.00886 (2025). arXiv:2506.00886 doi:10.48550/ARXIV.2506.00886

[65] Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. 2025. OTC: Optimal Tool Calls via Reinforcement Learning. *arXiv preprint arXiv:2504.14870* (2025).

[66] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[67] Yifan Wei, Xiaoyan Yu, Yixuan Weng, Tengfei Pan, Angsheng Li, and Li Du. 2025. AutoTIR: Autonomous Tools Integrated Reasoning via Reinforcement Learning. *CoRR* abs/2507.21836 (2025). arXiv:2507.21836 doi:10.48550/ARXIV.2507.21836

[68] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. WebDancer: Towards Autonomous Information Seeking Agency. arXiv:2505.22648 [cs.CL] https://arxiv.org/abs/2505.22648

[69] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. WebWalker: Benchmarking LLMs in Web Traversal. *CoRR* abs/2501.07572 (2025). arXiv:2501.07572 doi:10.48550/ARXIV.2501.07572

[70] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. 2025. Agentic Reasoning: Reasoning LLMs with Tools for the Deep Research. *CoRR* abs/2502.04644 (2025). arXiv:2502.04644 doi:10.48550/ARXIV.2502.04644

[71] Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. 2025. ReSum: Unlocking Long-Horizon Search Intelligence via Context Summarization. *arXiv preprint arXiv:2509.13313* (2025).

[72] Yang Xiao, Mohan Jiang, Jie Sun, Keyu Li, Jifan Lin, Yumin Zhuang, Ji Zeng, Shijie Xia, Qishuo Hua, Xuefeng Li, et al. 2025. LIMI: Less is More for Agency. *arXiv preprint arXiv:2509.17567* (2025).

[73] Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2025. SimpleTIR: End-to-End Reinforcement Learning for Multi-Turn Tool-Integrated Reasoning. *arXiv preprint arXiv:2509.02479* (2025).

[74] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*. Association for Computational Linguistics, Brussels, Belgium, 2369–2380. doi:10.18653/v1/D18-1259

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

[76] Dian Yu, Baolin Peng, Ye Tian, Linfeng Song, Haitao Mi, and Dong Yu. 2024. SIaM: Self-Improving Code-Assisted Mathematical Reasoning of Large Language Models. *CoRR* abs/2408.15565 (2024). arXiv:2408.15565 doi:10.48550/ARXIV.2408.15565

[77] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models. *CoRR* abs/2308.01825 (2023). arXiv:2308.01825 doi:10.48550/ARXIV.2308.01825

[78] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. *CoRR* abs/2503.18892 (2025). arXiv:2503.18892 doi:10.48550/ARXIV.2503.18892

[79] Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, and Fei Huang. 2025. EvolveSearch: An Iterative Self-Evolving Search Agent. *CoRR* abs/2505.22501 (2025). arXiv:2505.22501 doi:10.48550/ARXIV.2505.22501

[80] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547* (2025).

[81] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments. *arXiv preprint arXiv:2504.03160* (2025).

[82] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Bangkok, Thailand. http://arxiv.org/abs/2403.13372

[83] Yuanchen Zhou, Shuo Jiang, Jie Zhu, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. 2025. Fin-PRM: A Domain-Specialized Process Reward Model for Financial Reasoning in Large Language Models. *CoRR* abs/2508.15202 (2025). arXiv:2508.15202 doi:10.48550/ARXIV.2508.15202