

BENEFICIAL REASONING BEHAVIORS IN AGENTIC SEARCH AND EFFECTIVE POST-TRAINING TO OBTAIN THEM

Anonymous authors

Paper under double-blind review

ABSTRACT

Agentic search leverages large language models (LLMs) to interpret complex user information needs and execute a multi-step process of planning queries, executing searches, and adapting its strategy based on retrieved documents. This introduces unique challenges for LLMs’ reasoning and agentic capabilities in interacting with retrieval systems and the broader web. In this paper, we design a reasoning-driven LLM-based pipeline to study effective behaviors in agentic search. Using this pipeline, we analyze successful agentic search trajectories and extract four beneficial reasoning behaviors: Information Verification, Authority Evaluation, Adaptive Search, and Error Recovery. We then propose a new behavior priming technique to train more effective agentic search models. It synthesizes agentic search trajectories with target behaviors and integrates them into the agentic search model through supervised finetuning (SFT), followed by standard reinforcement learning (RL). Experiments on three benchmarks—GAIA, WebWalker, and HLE—demonstrate that behavior priming significantly improves agentic search accuracy, yielding over 35% gains in Llama3.2-3B and Qwen3-1.7B after RL. Ablation studies further show that priming effective behaviors is more beneficial than emphasizing result correctness: finetuning on trajectories with desirable reasoning behaviors but incorrect results outperforms training on trajectories with correct answers. Additional analysis reveals that reasoning patterns introduced by behavior priming provide a strong foundation for RL, enabling more effective exploration (higher pass@k and entropy) and test-time scaling (longer trajectories). Our code and models will be released as open source.

1 INTRODUCTION

Agentic search (Jin et al., 2025; Zheng et al., 2025; Li et al., 2025a; Moonshot AI, 2025) represents a new search paradigm in which search systems leverage large language models (LLMs) to perform multiple steps of agentic actions for complex user information needs. This includes decomposing tasks into sub-queries, planning search strategies, invoking search tools, analyzing results, and adapting strategies based on search results. Commercial systems implementing such techniques, such as ChatGPT’s Deep Research and Google’s AI Mode, have significantly enhanced user experiences and rapidly gained adoption (Zhou & Li, 2024; Business Insider, 2025; Verge, 2025).

The core enabler of agentic search is the reasoning and agentic capacity of LLMs, developed through large-scale post-training on single-step reasoning-oriented tasks such as mathematics, general problem-solving, and programming (DeepSeek-AI et al., 2025). The reasoning abilities demonstrated by frontier models are often tailored to their reinforcement learning (RL) scenarios—for instance, verification, subgoal setting, backtracking, and backward chaining for math Gandhi et al. (2025). However, their effectiveness in addressing challenges unique to agentic search—such as handling noisy results or adapting strategies based on incomplete information—remains uncertain.

In this paper, we first design an automatic LLM-based pipeline to study effective reasoning behaviors in agentic search. We collect the agentic search trajectories from multiple LLMs, and employ a reasoning LLM to analyze them, extracting behaviors that distinguish effective trajectories from ineffective ones. Through this process, we identify four reasoning behaviors critical for agentic search: **Information Verification** (validating results across sources), **Authority Evaluation** (assessing reliability and resolving conflicts), **Adaptive Search** (modifying strategies dynamically), and **Error Recovery** (detecting and correcting mistakes). The first two behaviors support reliable retrieval, while the latter two enable robust multi-step planning. Their presence strongly correlates with LLMs’ performance in agentic search tasks.

Building on these findings, we propose Behavior Priming, a technique that systematically instills these beneficial reasoning behaviors into agentic search models. The approach first curates a collection of trajectories that demonstrate the target behaviors by synthesizing and selecting from a large corpus of trajectories generated by LLMs. These curated trajectories serve as supervision data for fine-tuning models, explicitly injecting reasoning behaviors. This

behavior-prime phase provides the LLMs a solid foundation, enabling more effective exploration in the subsequent reinforcement learning phase. After this, we investigated the impact of these behaviors on RL by training models with and without behavior priming using a standard reinforcement learning setup.

Experiment results demonstrate that Behavior Priming leads to significantly greater performance gains for the final agentic search model (trained with SFT and RL). For instance, it boosted the final average performance of both Qwen3-1.7B and Llama-3.2-3B-Instruct by over 35% across three benchmarks (GAIA (Mialon et al., 2023), Web-WalkerQA (Wu et al., 2025c), and HLE (Phan et al., 2025a)) compared to applying direct RL on the non-primed models. Furthermore, our experiments reveal that: 1) Selecting SFT trajectories based on the presence of behaviors leads to better performance in RL than the standard approach of selecting based on final answer correctness. 2) Compared to SFT on trajectories with correct final answers, SFT on trajectories exhibiting these behaviors but yielding incorrect final answers results in suboptimal initial performance, but ultimately achieves higher performance after RL. These findings strongly highlights the primacy of reasoning behavior over outcome correctness for unlocking a model’s potential in RL.

Our analysis further clarifies the underlying mechanism for these improvements. During the Behavior Priming phase, the frequency of all four behaviors in the model’s reasoning process increases significantly, accompanied by a notable rise in both pass@k accuracy and the average number of steps per trajectory. This establishes a robust foundation for exploration and effective test-time scaling for the subsequent RL training. During the RL phase, behavior-primed models maintain a high level of policy entropy, whereas models without priming start with lower entropy that declines steeply, leading to premature policy convergence. Moreover, these non-primed models fail to cultivate the essential behaviors during RL endogenously.

In summary, our key contributions are as follows:

1. We identify four beneficial reasoning behaviors for agentic search by analyzing successful agentic trajectories with an LLM-based pipeline.
2. We propose Behavior Priming, a method that instills these beneficial behaviors into models via SFT to prepare them for subsequent RL training better.
3. We empirically demonstrate that Behavior Priming significantly unlocks a model’s potential in RL, enabling higher final performance by establishing a robust foundation for exploration and test-time scaling capabilities.

2 RELATED WORK

Agentic search is an emerging search paradigm where LLM-based systems autonomously and iteratively use web-related tools to gather external information for solving complex, fact-seeking tasks Xu & Peng (2025); OpenAI (2024); Anthropic (2025). The development of agentic search systems can be broadly categorized into two approaches. The first involves multi-agent collaboration systems Alzubi et al. (2025); GPTResearcher (2025); Li et al. (2025c); Zhang et al. (2025) within a meticulously designed, pre-defined workflow. The second approach focuses on single-agent, end-to-end systems Zheng et al. (2025); Jin et al. (2025); Nguyen et al. (2025) where a single underlying LLM iteratively invokes web search-related tools based on the context of previous steps. Research on agentic search training has predominantly concentrated on the latter one for its simplicity.

Influenced by the success of Reinforcement Learning in reasoning tasks like mathematics DeepSeek-AI et al. (2025); Jaech et al. (2024), many studies have adopted Reinforcement Learning with Verifiable Rewards (RLVR) for agentic search training Jin et al. (2025); Zheng et al. (2025); Li et al. (2025a); Moonshot AI (2025). However, due to the scarcity of data that contains tool use and real-world interaction, the foundational capabilities of LLMs for agentic tasks are relatively underdeveloped Tao et al. (2025); Li et al. (2025b); Shi et al. (2025). Consequently, a common way for training search agents with RLVR require a ”cold-start” instruction-tuning phase Tao et al. (2025); Li et al. (2025a); Jie Ouyang (2025); Moonshot AI (2025) before applying RL to familiarize the model with the task. However, the specific abilities instilled during this phase and their impact on the subsequent RL stage requires further investigation.

RLVR has emerged as a powerful paradigm for enhancing complex reasoning in language models, with the notable breakthrough of Deepseek-R1 DeepSeek-AI et al. (2025) demonstrating its ability to yield significant improvements and the emergence of in-context reasoning behaviors. However, the emergence of reasoning capabilities in the RL process is not guaranteed Yeo et al. (2025). Certain models, notably the Qwen 2.5 series Yang et al. (2025), have been widely demonstrated to achieve significant improvements with RL. In contrast, other models fail to show similar improvement. Recent works Yeo et al. (2025); Wu et al. (2025a); Liu et al. (2025); Setlur et al. (2025) have shown that the base model’s ability to explore and extrapolate computation at test-time is pivotal for effective improvement in RL training. Models that benefit significantly from RL usually already exhibit some key behaviors such as self-

108 reflection and verification before the RL training Gandhi et al. (2025); Liu et al. (2025); Yeo et al. (2025). Furthermore,
 109 instilling these abilities through supervised-finetuning on models that initially lack them is effective in raising their
 110 performance ceiling in RL Yeo et al. (2025); Liu et al. (2025); Gandhi et al. (2025). However, relevant research has
 111 been predominantly concentrated on the mathematical domain. For broader applications, such as real-world agentic
 112 tasks, the critical capabilities required for successful RL, and the methods to acquire them, remain open questions. In
 113 this paper, we explore the nature of this capability gap by identifying the essential capabilities for agentic search and
 114 exploring methods to obtain them.

116 3 IDENTIFYING BENEFICIAL BEHAVIORS IN AGENTIC SEARCH

117
 118 To identify beneficial behaviors in agentic search, we first develop a standard agentic search framework to enable
 119 our study across different LLMs, and then design an LLM-based analytical pipeline to systematically discover the key
 120 reasoning patterns from agentic search trajectories. We also validate the importance of these behaviors by investigating
 121 the correlation between their frequency in trajectories from various LLMs and the final task performance.

123 3.1 STANDARD AGENTIC SEARCH FRAMEWORK

124
 125 To facilitate our study across different LLMs, we simplified the agent framework in (Chandrasekhar et al., 2025) to
 126 build a standard end-to-end agentic search framework that easily integrates various LLMs as the underlying model,
 127 with prompts shown in Appendix A.1. This standardized approach ensures our analysis focuses on the model’s core
 128 capabilities rather than framework-specific artifacts. This framework operates in an iterative process: at step k , the
 129 model receives an input x_k , which consists of the initial user query q and the accumulated history context ctx_k . It then
 130 produces an output $y_k = \langle t_k, a_k \rangle$, where t_k is the thinking process and a_k is the action. The entire interaction is
 131 captured as a trajectory $\mathcal{T} = (\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_n, y_n \rangle)$.

132 The model’s action a_k at each step is selected from a predefined set of three actions. Each action has a distinct effect
 133 on the iterative process:

- 134 • **search**: The model queries an external search API and receives retrieved documents as an observation,
 135 obs_k .
- 136 • **summary**: The model condenses the current context to manage context length. This summary will replace
 137 the entire context for the next step.
- 138 • **answer**: The process terminates, and a_k is provided as the final answer.

139
 140 The history context ctx_k serves as the model’s memory, providing the necessary context from previous steps. After
 141 step k , the history context is updated based on the action taken, preparing the model for the next step:

$$142 \quad ctx_{k+1} = \begin{cases} ctx_k + y_k + obs_k, & \text{if } a_k = \text{search} \\ a_k, & \text{if } a_k = \text{summary} \end{cases}$$

146 3.2 IDENTIFY BENEFICIAL REASONING BEHAVIORS FOR AGENTIC SEARCH

147
 148 To identify beneficial reasoning behaviors, we leveraged our standardized agent framework to conduct a comparative
 149 analysis of model trajectories. Specifically, we integrated a strong model (Gemini 2.5 Flash) and a weak model
 150 (Qwen3-1.7B) into the framework, and tasked them with answering an identical set of questions. After evaluation,
 151 we extract a set of questions where Gemini 2.5 Flash succeeded but Qwen3-1.7B failed. This approach allows for a
 152 controlled comparison of the models’ reasoning behaviors when faced with the same problem and equipped with the
 153 same tools, thereby isolating the key differences that lead to divergent outcomes.

154 Inspired by an automated rule generation method Wang & Xiong (2025), we developed an automatic LLM-based
 155 pipeline to identify effective reasoning behaviors that lead to the successful completion of agentic tasks. Prompts can
 156 be found in Appendix A.2. Our process involved three main steps. First, we prompted a reasoning LLM to analyze pairs
 157 of successful and failed trajectories for individual questions. The LLM was provided with the initial question and both
 158 complete trajectories, then instructed to determine why one attempt succeeded while the other failed. Second, we gath-
 159 ered the LLM’s analyses for 200 randomly selected trajectory pairs and prompted it to summarize the key reasoning
 160 behaviors observed in the successful trajectories. Finally, we tasked the LLM with consolidating and deduplicating
 161 these summaries to produce the final set of common behaviors. We then manually reviewed these behaviors to confirm
 that the identified behaviors were indeed recurring patterns across multiple tasks.

Through this analysis pipeline, we identified four essential reasoning behaviors for agentic search. These can be divided into two main categories: the first two focus on *information retrieval*. At the same time, the latter two are more general strategies for the *multi-step agentic planning* ability. These behaviors are: **Information Verification**, which involves validating search results across multiple sources, performing cross-referencing, and explicitly citing evidence in the reasoning process; **Authority Evaluation**, the ability to identify conflicts among different search results and analyze source credibility to prioritize the most authoritative information; **Adaptive Search**, the practice of dynamically modifying search strategies based on previous outcomes; and **Error Recovery**, the capacity for recognizing and correcting mistakes made in prior steps. The following box provides examples of these behaviors in Gemini-2.5-Flash’s trajectories.

Examples of four behaviors

Information Verification — Cross-source confirmation and exact-quote checking:

“My task is clear: verify if the quoted text exactly matches Greetham’s article... accuracy is paramount; I’ll use ‘uncoupled’, ‘authors’, ‘mis-transmission’, and ‘veil’ to zero in on the relevant section.”

Authority Evaluation — Narrowed focus to official data source:

“I’m aiming for the USGS’s own reports or databases, like the ‘Nonindigenous Aquatic Species’ page, to get the most reliable data.”

Adaptive Search — Methodical refinement based on previous results:

“The search engine may not have indexed the quote perfectly, or the user’s quote may differ slightly... I’ll refine my strategy.”

Error Recovery — Recognized previous incomplete results and corrected:

“My initial search didn’t deliver the goods... I need to get more precise.”

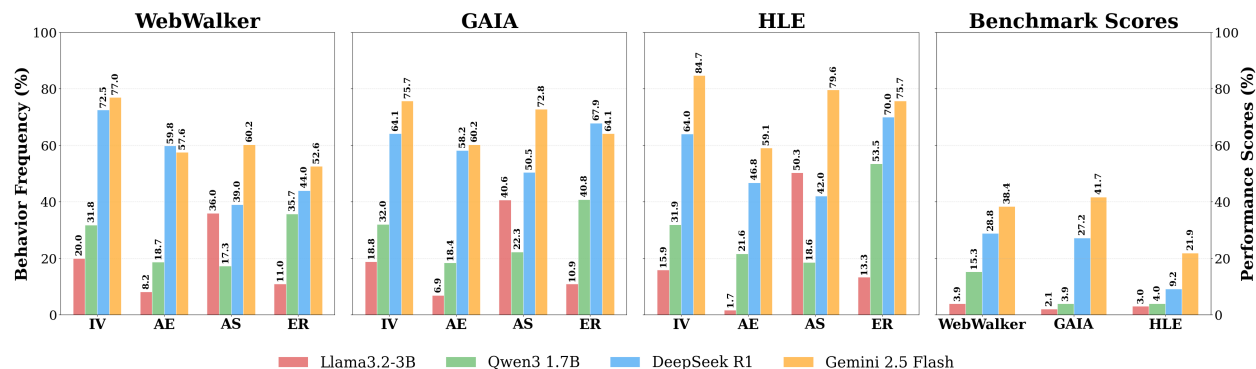


Figure 1: Comparison of different LLMs as the underlying model of our agent framework across four benchmarks. The first three plots represent the frequency of behaviors, and the last plot indicates scores on benchmarks. *Abbreviations: IV = Information Verification, AE = Authority Evaluation, AS = Adaptive Search, ER = Error Recovery.*

3.3 VALIDATING THE IMPORTANCE OF BENEFICIAL BEHAVIORS

We then developed an automatic framework to measure the frequency of these four behaviors in agent trajectories. Specifically, we provide the full trajectories of models to an LLM and prompt it as an automated evaluator to identify the presence of each behavior within these trajectories. Detail prompts are provided in Appendix A.3. We define *behavior frequency* as the proportion of trajectories that exhibit a specific behavior. Using this analysis framework, we evaluated Gemini 2.5 Flash, DeepSeekR1, Llama3.2-3B-Instruct, and Qwen3-1.7B on three widely accepted benchmarks for agentic search.

As illustrated in Figure 1, results across different models demonstrate a strong correlation between model performance and the frequency of these beneficial behaviors. The performance ranking of the four models directly corresponds to their ranking in behavior frequency, which validates the critical importance of these behaviors for successful agentic search.

4 BEHAVIOR PRIMING: UNLOCKING THE RL POTENTIAL OF SEARCH AGENTS

4.1 INSTILLING BEHAVIORS VIA SUPERVISED FINE-TUNING

Having established the essential behaviors for agentic search, we now investigate how the preceding SFT phase impacts a model’s performance during subsequent RL Learning. The conventional approach typically fine-tunes a model on trajectories from strong models for distillation, or filters for trajectories that yield a correct outcome. However, we hypothesize that a more effective strategy should focus on instilling the desired reasoning behaviors, regardless of the final answer, which we term “**Behavior Priming**”.

To test this hypothesis, we curated a suite of specialized datasets. First, we generated a large corpus of trajectories using Gemini 2.5 Flash. Each trajectory in this corpus was then analyzed for two criteria: the correctness of its final answer and the presence of the four beneficial behaviors. From this analyzed corpus, we filtered and created several equally-sized datasets: **1) SFT (Random)**: Trajectories randomly selected from the initial corpus, serving as an SFT distillation baseline. **2) SFT (Correct)**: Trajectories with a correct final answer, irrespective of the behaviors exhibited. **3) Behavior Prime**: Trajectories exhibiting all four beneficial behaviors, regardless of the final outcome. **4) Behavior Prime (Incorrect)**: Trajectories exhibiting all four behaviors but resulting in an incorrect final answer. **5) Behavior Prime (Correct)**: Trajectories exhibiting all four behaviors and resulting in a correct final answer.

While the **SFT (Random)** and **SFT (Correct)** datasets are standard practice, their trajectories do not necessarily contain our identified reasoning behaviors. Conversely, trajectories in the **Behavior Prime** dataset do not always yield the correct answer. To disentangle the influence of the reasoning process from the final outcome, we use the **Behavior Prime (Incorrect)** dataset to verify that the reasoning process itself, rather than a correct outcome, is the key to providing a strong foundation for subsequent RL training. In parallel, the **Behavior Prime (Correct)** dataset was created to assess the synergistic effect of combining a high-quality reasoning process with a successful outcome. Since a capable model generated the corpus, the **SFT (Random)** and **SFT (Correct)** datasets serve as strong baselines, allowing us to measure the added value of our targeted curation against previous simple distillation methods.

With the datasets prepared, we instill the target behaviors into a base model via supervised fine-tuning. We train models on multi-step trajectories by treating each step as an independent training sample. Formally, given a trajectory dataset $\mathcal{D}_{\mathcal{T}} = \{\mathcal{T}_i\}_{i=1}^N$, where each trajectory is a sequence $\mathcal{T}_i = (\langle x_1^i, y_1^i \rangle, \langle x_2^i, y_2^i \rangle, \dots, \langle x_{L_i}^i, y_{L_i}^i \rangle)$ with length L_i , the dataset for SFT \mathcal{D}_{SFT} is constructed by aggregating all input-output pairs from all trajectories:

$$\mathcal{D}_{SFT} = \{\langle x_k^i, y_k^i \rangle \mid \mathcal{T}_i \in \mathcal{D}_{\mathcal{T}}, 1 \leq k \leq L_i\}$$

Since we use each step as an individual training sample rather than concatenating the full trajectory, it is unnecessary to mask the loss on environment observations. The model π_{θ} is trained on \mathcal{D}_{SFT} with the standard loss function for autoregressive sequence generation:

$$\mathcal{L}_{SFT}(\theta) = \mathbb{E}_{\langle x, y \rangle \sim \mathcal{D}_{SFT}} \left[- \sum_{j=1}^{|y|} \log \pi_{\theta}(y_j \mid y_{< j}, x) \right]$$

4.2 REINFORCEMENT LEARNING

After the SFT phase, each of these SFT-checkpoints, along with the original base model (as a no-SFT baseline), undergoes the same RL training process for an identical number of steps. Similarly, in this RL phase, each step of a trajectory is treated as an independent training sample.

We optimize the policy π_{θ} using the GRPO Shao et al. (2024) algorithm. For each question q , we sample a set of G trajectories rollout $\{\mathcal{T}_i\}_{i=1}^G$. The policy is then updated by aggregating over all individual steps from the sampled trajectories. The loss function is defined as:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, \{\mathcal{T}_i\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot \mid q)} \left[\sum_{i=1}^G \sum_{k=1}^{L_i} \sum_{t=1}^{|y_k|} \frac{1}{|y_k|} \min \left(r_{i,k,t}(\theta) \hat{A}_i, \text{clip} \left(r_{i,k,t}(\theta), 1 \pm \varepsilon \right) \hat{A}_i \right) \right] \quad (1)$$

where $r_{i,k,t}(\theta)$ is the importance sampling ratio, x_k^i and y_k^i are the input and output at step k of trajectory i , and \hat{A}_i is the advantage estimate for that trajectory.

We employ an outcome-based, trajectory-level reward signal to guide the training. For each completed trajectory \mathcal{T}_i , an LLM-judge evaluates the final answer a_i against the ground-truth solution g , assigning a binary reward R_i of 1 for

Dataset	Information Verification	Authority Evaluation	Adaptive Search	Error Recovery	Outcome Accuracy	Avg. Steps / Traj.	# Traj.	# Total Steps
SFT (Random)	71.7	42.2	52.3	36.2	40.0%	4.6	4.3k	20k
SFT (Correct)	85.7	52.2	53.2	28.0	100.0%	3.9	5.1k	20k
Behavior Prime	100.0	100.0	100.0	100.0	49.8%	6.8	2.9k	20k
Behavior Prime (Incorrect)	100.0	100.0	100.0	100.0	0.0%	7.6	2.6k	20k
Behavior Prime (Correct)	100.0	100.0	100.0	100.0	100.0%	5.9	3.4k	20k

Table 1: The behavior frequencies, outcome accuracy, and trajectory statistics for SFT datasets.

a correct answer and zero otherwise. This reward R_i is used as reward for every step within that trajectory, and the advantage estimate \hat{A}_i is also constant for all steps, defined as $\hat{A}_i = \frac{R_i - \text{mean}(\{R_k\})}{\text{std}(\{R_k\})}$.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset We use the SFT and RL dataset from the web agent dataset of Li et al. (2025b). For SFT, we utilize only the questions and ground truth answers. We generated a large trajectory corpus by using Gemini 2.5 Flash as the underlying LLM in our agent framework, sampling 10 trajectories for each question, and randomly selecting data samples that meet specific criteria to create each SFT dataset. Detailed statistics for these datasets are presented in Table 1. For the RL phase, we utilized the complete 10427 instance in original dataset.

Training Details We adopt Qwen3-1.7B Yang et al. (2025) and Llama3.2-3B-Instruct Meta (2024) as our base models. For SFT, each model was fine-tuned for three epochs with a batch size of 8. The RL training was conducted on the verl-agent framework Feng et al. (2025) for 300 steps, with a batch size of 32 and a group size of 8 for GRPO.

Benchmarks and Evaluation We evaluate on three widely-accepted benchmarks for search agents: WebWalkerQA Wu et al. (2025c), GAIA Mialon et al. (2023), and Humanity’s Exam (HLE) Phan et al. (2025b). For the GAIA benchmark, we follow prior work Li et al. (2025c); Wu et al. (2025b); Li et al. (2025b) and use the subset of 103 text-based examples. To assess performance, we adopt an LLM-as-Judge approach, using GPT-4o-mini to score the final answers. During this evaluation, each trajectory is limited to a maximum of 25 steps. For our main evaluation, the model’s temperature is set to 0.0, whereas for the pass@k evaluation, it is set to 1.0.

5.2 MAIN RESULTS

Behavior Priming Effectively Increases Headroom for RL As shown in Table 2, models that first undergo behavior priming (trained on the **Behavior Prime**, **Behavior Prime (Incorrect)**, or **Behavior Prime (Correct)** datasets) consistently achieve significantly larger performance gains during the subsequent RL phase compared to models trained with RL directly. For instance, the final scores of both Llama-3.2-Instruct and Qwen3-1.7B improved by over 35% relative to applying RL to the non-primed base models (19.3 vs. 14.3 and 22.3 vs. 13.9, respectively). Furthermore, Behavior Priming significantly outperforms the simple distillation baseline (trained on the **SFT (random)** dataset), which demonstrates that our targeted data curation method is effective at instilling beneficial properties for subsequent RL training. These results demonstrate that by establishing a robust foundation through behavior-centric SFT, Behavior Priming creates greater “headroom” for improvement during RL, ultimately leading to superior final performance.

Reasoning Behavior is More Important than Outcome Correctness While fine-tuning on the **SFT (Correct)** dataset yields the best scores immediately after the SFT phase, this advantage does not lead to the best ultimate performance. During the subsequent RL training, the model trained on the **SFT (Correct)** dataset is surpassed by those trained with our behavior priming method (trained on the **Behavior Prime**, **Behavior Prime (Incorrect)**, or **Behavior Prime (Correct)** datasets). This finding strongly suggests that although outcome correctness provides a powerful initial boost, focusing on reasoning behaviors lays a more solid foundation for RL. Priming the model on behaviors provides it with greater headroom for improvement, which is the key to achieving superior final performance.

Furthermore, this conclusion is powerfully supported by the results from the **Behavior Prime (Incorrect)** dataset. Despite achieving suboptimal improvement after SFT, this model ultimately acquired the largest gains during RL training and even achieved the strongest final performance. This result provides strong evidence that a model can still

Table 2: Overall performance (*in accuracy %*) of Qwen3-1.7B and Llama3.2-3B-Instruct before and after RL fine-tuning. Bold numbers indicate the maximum score.

Method	GAIA				WebWalker	HLE	Overall
	Level 1	Level 2	Level 3	Avg.	Avg.	Avg.	Avg.
<i>Before RL</i>							
Qwen3-1.7B	7.7	1.9	0.0	3.9	15.3	4.0	7.7
Qwen3-1.7B + SFT (Random)	12.8	7.7	0.0	8.7	23.5	5.4	12.5
Qwen3-1.7B + SFT (Correct)	10.3	9.6	0.0	8.7	24.3	4.8	12.6
Qwen3-1.7B + Behavior Prime	12.8	7.7	0.0	8.7	22.0	4.6	11.8
Qwen3-1.7B + Behavior Prime (Incorrect)	7.7	11.5	0.0	8.7	15.3	3.5	9.3
Qwen3-1.7B + Behavior Prime (Correct)	10.3	9.6	0.0	8.7	19.1	6.2	11.2
Llama3.2-3B-Instruct	2.6	3.8	0.0	3.0	3.9	2.1	3.0
Llama3.2-3B-Instruct + Behavior Prime	20.5	9.6	0.0	12.6	26.7	5.2	14.2
<i>After RL</i>							
Qwen3-1.7B + RL	15.4	11.5	0.0	11.7	26.1	3.9	13.9
Qwen3-1.7B + SFT (Random) + RL	18.0	11.5	16.7	14.6	33.2	7.4	18.4
Qwen3-1.7B + SFT (Correct) + RL	23.1	17.3	0.0	17.5	36.8	5.8	20.1
Qwen3-1.7B + Behavior Prime + RL	28.2	21.2	0.0	21.4	37.2	7.8	22.3
Qwen3-1.7B + Behavior Prime (Incorrect) + RL	30.8	26.9	8.3	27.2	35.5	7.8	23.5
Qwen3-1.7B + Behavior Prime (Correct) + RL	30.8	23.1	16.7	25.2	37.8	7.8	23.6
Llama3.2-3B-Instruct + RL	12.8	11.5	8.3	11.7	23.8	7.4	14.3
Llama3.2-3B-Instruct + Behavior Prime + RL	25.6	11.5	8.3	16.5	34.0	7.5	19.3

acquire beneficial reasoning processes from trajectories that all lead to incorrect outcomes. It demonstrates that the reasoning process itself, independent of the outcome, is the most critical factor for building a strong foundation for subsequent RL training.

5.3 TRAINING DYNAMICS

To better understand the effect of Behavior Priming, we analyze the model’s properties during both the supervised fine-tuning phase and the subsequent RL training. Our analysis reveals that priming instills essential capabilities for exploration and extrapolation of computation at test time, which unlock greater potential for the RL phase.

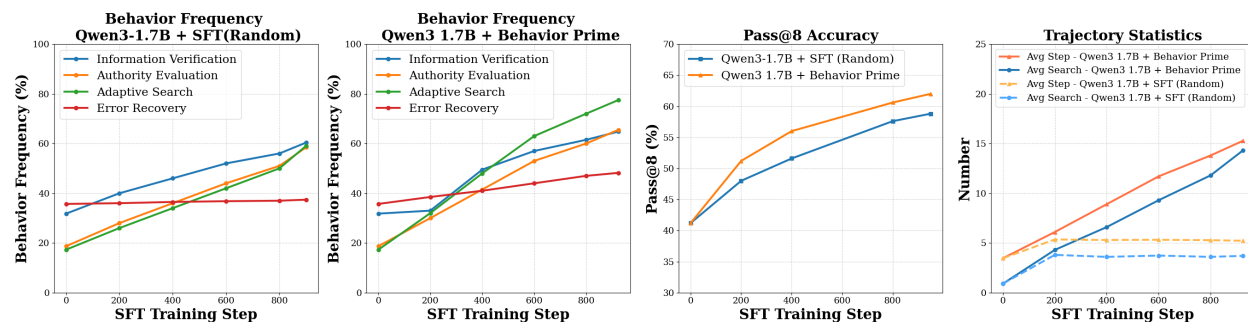


Figure 2: Qwen3 1.7B + SFT (Random) and Qwen3 1.7B + Behavior Prime’s behavior frequencies, Pass@8 accuracy, and trajectories statistics (average step number and average search action number per trajectory) on the WebWalkerQA benchmark with different checkpoints during the supervised fine-tuning process.

Behavior Priming Encourages Exploration We first examine how model capabilities evolve during the behavior priming process. We monitored the model’s Pass@8 performance, average trajectory step length, and the frequency of the four beneficial behaviors as SFT progressed on the SFT (Random) and Behaviors Prime datasets as shown in Figure 2. In contrast, all metrics show an upward trend on both datasets; the growth is substantially more pronounced when training on the Behavior Prime dataset, with the frequencies of all four behaviors, the model’s Pass@8 performance, and the average number of steps per trajectory all increasing to a much greater extent. This indicates that the model is learning to explore more diverse paths and allocate more test-time compute resources by performing more steps. This process equips the model with crucial exploration abilities, laying a stronger foundation for the RL stage.

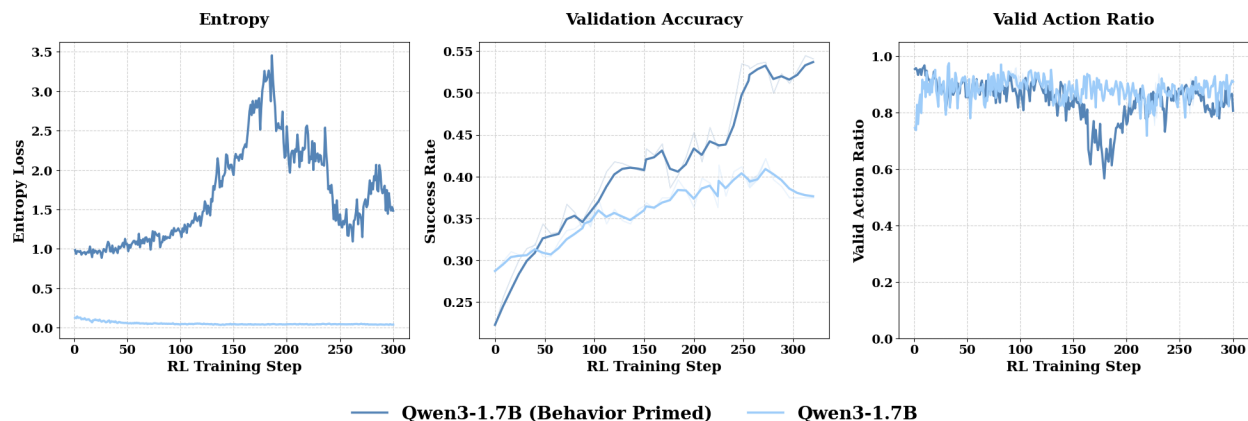


Figure 3: The entropy loss, validation accuracy, and valid action ratio trend during the RL process of Qwen3-1.7B and Qwen3-1.7B with behavior priming (SFT on the **Behavior Prime** dataset). The Valid Action Ratio is the percentage of steps in which the model generates a syntactically valid action.

Sustained Exploration Translates to a Higher RL Ceiling We also analyzed the training dynamics during the RL phase. We compared the policy entropy and validation performance of the model with behavior priming against those of the initial base model, which was trained without behavior priming (from scratch using RL). Results demonstrated that the behavior-primed model consistently maintains a high policy entropy throughout RL training, showing little tendency to collapse. In contrast, the base model begins with a lower entropy that rapidly collapses. This high entropy signifies that the primed model sustains a richer exploratory ability, continuously seeking diverse strategies rather than prematurely converging to a suboptimal policy.

This sustained exploration translates directly to performance. The validation curve shows that while the base model converges more quickly to a low plateau, the behavior-primed model converges more slowly and reaches a higher performance ceiling. This confirms that Behavior Priming creates more headroom for improvement, enabling the RL algorithm to discover more effective and robust policies.

Disentangling Reasoning Behaviors from Format Learning Notably, our results demonstrate that unfamiliarity with the task’s required output format is not the primary barrier preventing the base model from improving during RL. As shown in Figure 3, although the base Qwen3-1.7B model initially has a slightly lower valid action ratio (the percentage of steps in which the model generates a syntactically valid action) than the primed model, it successfully masters the action format within just 20 steps and maintains a high ratio thereafter. In contrast, the valid action ratio for the behavior-primed model is even less stable. This phenomenon provides strong evidence that the performance gains from Behavior Priming stem from the targeted reasoning behaviors themselves, rather than from simple format familiarization (e.g., learning tool-use syntax).

5.4 EFFECT OF SFT DATA SIZE ON BEHAVIOR PRIMING

We also investigated how the scale of the SFT data affects the efficacy of Behavior Priming. To this end, we fine-tuned Qwen3-1.7B on subsets of our **Behavior Prime** dataset of varying sizes: 5k, 10k, and the full 20k samples. We then subjected each of these behavior-primed models to RL training and compared their final performance against a baseline that received no behavior-priming process (denoted as 0k).

Results in Figure 4 reveal a clear scaling trend: as the size of the SFT dataset increases, the model’s final performance after RL consistently improves. This demonstrates that the benefits of Behavior Priming is scalable, with performance ceiling increasing by leveraging larger-scale data for the initial priming phase.

6 DISCUSSION: CAN PROCESS REWARDS INDUCE EMERGENT OF BEHAVIORS?

We observed that the base model fails to spontaneously acquire these reasoning behaviors during direct RL training (See the behavior frequency in Table 3). This inspires us to investigate whether we could guide the emergence of these behaviors with a process reward signal during the RL phase, instead of directly instilling them in a preliminary

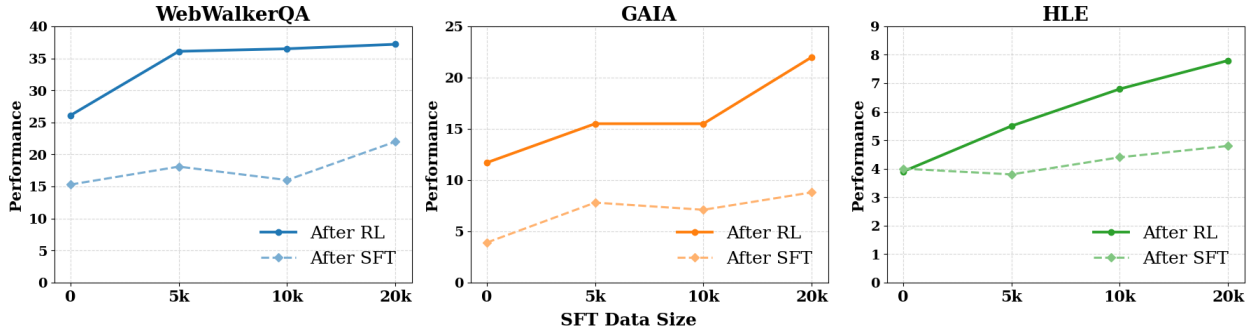


Figure 4: Qwen3-1.7B’s performance across three benchmarks after Behavior Priming with different dataset sizes, and corresponding performance after the subsequent RL training.

SFT phase. Specifically, we combined the standard outcome-based reward with a process reward that encourages the exhibition of reasoning behaviors in the model’s trajectory. For a rollout trajectory \mathcal{T}_i generated during RL, we use the analysis framework in Section 4.1 to identify behaviors in this trajectory, and define the final reward R_i as:

$$R_i = R_{\text{outcome}} + \beta \times N$$

where N is the number of reasoning behaviors present in the trajectory and β is a weighting hyperparameter, which we set to 0.1.

As shown in Table 3, this reward-shaping approach successfully increases the frequency of all four behaviors. However, this increase did not translate into better performance. On the contrary, the final task scores were even worse compared to standard RL with an outcome-only reward. This result aligns with the findings of previous work on math and code reasoning DeepSeek-AI et al. (2025), suggesting that the model learns to “reward hack”—it mimics the surface-level patterns of the behaviors to maximize the process reward but fails to grasp their functional essence for effective problem-solving. This finding reveals that our SFT-based Behavior Priming method is a more effective way to improve model performance, which instills a deeper, more grounded understanding of the reasoning process that direct reward shaping cannot achieve.

Table 3: Results of behavior frequency (averages across benchmarks, with increments relative to Qwen3-1.7B before RL training) and downstream performance evaluation for standard RL and behavior guided RL with process rewards.

Model	Behavior Frequency				Performance		
	Information Verification	Authority Evaluation	Adaptive Search	Error Recovery	WebWalkerQA	GAIA	HLE
Qwen3-1.7B	28.3	18.1	18.2	39.7	15.3	3.9	4.0
Qwen3-1.7B + RL	25.2 (-3.1)	9.8 (-8.3)	21.5 (+3.3)	11.2 (-28.5)	26.1	16.3	3.9
Qwen3-1.7B + behavior guided RL	67.3 (+39.0)	50.8 (+32.7)	82.8 (+64.6)	86.5 (+46.8)	15.4	7.8	4.4

7 CONCLUSION

In this work, we address the critical question of how to best prepare LLMs for Reinforcement Learning in agentic search. Unlike in domains such as mathematics, foundational reasoning capabilities for these real-world agentic tasks are often underdeveloped, making this preparatory stage particularly important. To this end, we identified four essential reasoning behaviors for agentic search: Information Verification, Authority Evaluation, Adaptive Search, and Error Recovery. We then proposed Behavior Priming, a method that uses supervised fine-tuning to instill these behaviors before the RL stage. Our findings reveal that the presence of these reasoning behaviors within SFT trajectory data is more important than the correctness of the final outcome. This demonstrates that Behavior Priming builds a strong reasoning foundation for unlocking a model’s potential for self-improvement during RL training. Our analysis reveals that this method fosters a robust exploratory capability that translates to sustained high policy entropy during RL, preventing premature policy collapse. This work highlights a promising direction for exploring how targeted post-training can systematically unlock the reinforcement learning potential to build more capable and robust agents equipped to handle complex real-world tasks.

8 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we will make our source code publicly available upon publication. The code will include scripts for model training, data processing, and evaluation. We provide details of dataset, evaluation and training in Section 5.1. We also provide all prompts used in our method in Appendix A.

REFERENCES

- Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, et al. Open deep search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*, 2025.
- Anthropic. Claude can now search the web, 2025. URL <https://www.anthropic.com/news/web-search>.
- Business Insider. Apple and google disagree on ai cutting into search. *Business Insider*, May 2025. URL https://www.businessinsider.com/apple-google-disagree-ai-cutting-into-search-2025-5?utm_source=chatgpt.com. Accessed: 2025-09-22.
- Prahaladh Chandrahasan, Jiahe Jin, Zhihan Zhang, Tevin Wang, Andy Tang, Lucy Mo, Morteza Ziyadi, Leonardo F. R. Ribeiro, Zimeng Qiu, Markus Dreyer, Akari Asai, and Chenyan Xiong. Deep research comparator: A platform for fine-grained human annotations of deep research agents, 2025. URL <https://arxiv.org/abs/2507.05495>.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Gao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL <https://arxiv.org/abs/2505.10978>.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.
- GPTResearcher. Say goodbye to hours of research, 2025. URL <https://gptr.dev/>.
- 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.

- 540 Yucong Luo Mingyue Cheng Qi Liu Zirui Liu Shuo Yu Daoyu Wang Jie Ouyang, Ruiran Yan. Training power-
541 ful llm agents with end-to-end reinforcement learning, 2025. URL [https://github.com/0russwest0/](https://github.com/0russwest0/Agent-R1)
542 Agent-R1.
- 543 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han.
544 Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint*
545 *arXiv:2503.09516*, 2025.
- 547 Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zheng-
548 wei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint*
549 *arXiv:2507.02592*, 2025a.
- 550 Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen,
551 Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distilla-
552 tion and agentic rl. *arXiv preprint arXiv:2508.13167*, 2025b.
- 553 Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou.
554 Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*,
555 2025c.
- 557 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understand-
558 ing r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- 559 Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL <https://ai.meta.com/blog/meta-llama-3/>.
- 562 Grégoire Mialon, Clémentine Fourier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for
563 general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.
- 564 Moonshot AI. End-to-end rl training for emerging agentic capabilities, 2025. URL <https://moonshotai.github.io/Kimi-Researcher/>.
- 567 Xuan-Phi Nguyen, Shrey Pandit, Revanth Gangi Reddy, Austin Xu, Silvio Savarese, Caiming Xiong, and Shafiq
568 Joty. Sfr-deepresearch: Towards effective reinforcement learning for autonomously reasoning single agents. *arXiv*
569 *preprint arXiv:2509.06283*, 2025.
- 570 OpenAI. Introducing chatgpt search, 2024. URL <https://openai.com/index/introducing-chatgpt-search/>.
- 572 Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed
573 Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint arXiv:2501.14249*, 2025a.
- 574 Long Phan et al. Humanity’s Last Exam. *ArXiv*, abs/2501.14249, 2025b.
- 576 Amrith Setlur, Matthew YR Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and
577 Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint*
578 *arXiv:2506.09026*, 2025.
- 580 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang,
581 YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.
582 *arXiv preprint arXiv:2402.03300*, 2024.
- 583 Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin,
584 King Zhu, Minghao Liu, Jian Yang, Ge Zhang, Jiaheng Liu, Changwang Zhang, Jun Wang, Yuchen Eleanor Jiang,
585 and Wangchunshu Zhou. Taskcraft: Automated generation of agentic tasks, 2025. URL <https://arxiv.org/abs/2506.10055>.
- 587 Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu
588 Wang, Yong Jiang, et al. Webshaper: Agentic data synthesizing via information-seeking formalization. *arXiv*
589 *preprint arXiv:2507.15061*, 2025.
- 591 Verge. Google searches are falling in safari for the first time ever — probably because
592 of ai. *The Verge*, May 2025. URL [https://www.theverge.com/news/662725/](https://www.theverge.com/news/662725/google-search-safari-ai-apple-eddy-cue-testimony?utm_source=chatgpt.com)
593 [google-search-safari-ai-apple-eddy-cue-testimony?utm_source=chatgpt.com](https://www.theverge.com/news/662725/google-search-safari-ai-apple-eddy-cue-testimony?utm_source=chatgpt.com).
Accessed: 2025-09-22.

- 594 Tevin Wang and Chenyan Xiong. Autorule: Reasoning chain-of-thought extracted rule-based rewards improve pref-
595 erence learning, 2025. URL <https://arxiv.org/abs/2506.15651>.
- 596 Haoze Wu, Cheng Wang, Wenshuo Zhao, and Junxian He. Model-task alignment drives distinct rl outcomes. *arXiv*
597 *preprint arXiv:2508.21188*, 2025a.
- 598 Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi,
599 Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint*
600 *arXiv:2505.22648*, 2025b.
- 601 Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu
602 Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025c. URL <https://arxiv.org/abs/2501.07572>.
- 603 Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications.
604 *arXiv preprint arXiv:2506.12594*, 2025.
- 605 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen
606 Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- 607 Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reason-
608 ing in llms. *arXiv preprint arXiv:2502.03373*, 2025.
- 609 Wentao Zhang, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. Agentorchestra: A hierarchical
610 multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025.
- 611 Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepRe-
612 searcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments. *ArXiv*, abs/2504.03160,
613 2025.
- 614 Tao Zhou and Songtao Li. Understanding user switch of information seeking: From search engines to generative ai.
615 *Journal of librarianship and information science*, pp. 09610006241244800, 2024.
- 616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647

A PROMPTS

A.1 PROMPT FOR AGENT FRAMEWORK

We support the integration of both models with internal thinking (like Gemini 2.5 series, Qwen3 series, DeepSeek R1) and models without internal thinking. For models without internal thinking, we explicitly prompt them first to generate a thinking process and then the action. The prompts for both models as shown below:

Agent Framework Prompt for Models with Internal Thinking

Your are a research assistant with the ability to perform web searches to answer questions. You can answer a question with many turns of search and reasoning.

Based on the history information, you need to suggest the next action to complete the task.

You will be provided with:

1. Your history search attempts: query in format `<search> query </search>` and the returned search results in `<information>` and `</information>`.
2. The question to answer.

IMPORTANT: You must strictly adhere to the following rules:

1. Choose **ONLY ONE** action from the list below for each response, **DO NOT** perform more than one action per step.
2. Follow the exact syntax format for the selected action, **DO NOT** create or use any actions other than those listed.
3. ****Don't do duplicate search.**** Pay attention to the history search results.

Valid actions:

1. `<search> query </search>`: search the web for information if you consider you lack some knowledge.
2. `<answer> answer </answer>`: output the final answer if you consider you are able to answer the question. The answer should be short and concise. No justification is needed.
3. `<summary> important parts of the history turns </summary>`: summarize the history turns. Reflect the search queries and search results in you history turns, and keep the information you consider important for answering the question and generating your report. Still keep the tag structure, keep search queries between `<search>` and `</search>`, and keep search results between `<information>` and `</information>`. The history turn information for your subsequent turns will be updated accoring to this summary action.

Format:

You should pay attention to the format of your output. You can choose ****ONLY ONE**** of the following actions:

- If You want to search, You should put the query between `<search>` and `</search>`.
 - If You want to summarize the history turns, You should put the summary between `<summary>` and `</summary>`.
 - If You want to give the final answer, You should put the answer between `<answer>` and `</answer>`.
- You can only use **ONE** action per response.

Note: text between `<information></information>` is the search results from search engine after you perform a search action, ****DO NOT**** include any information in `<information></information>` in your output.

Question: {question}

History Turns: (empty if this is the first turn)

Agent Framework Prompt for Models without Internal Thinking

Your are a research assistant with the ability to perform web searches to answer questions. You can answer a question with many turns of search and reasoning.

Based on the history information, you need to suggest the next action to complete the task.

You will be provided with:

1. Your history search attempts: query in format `<search> query </search>` and the returned search results in `<information>` and `</information>`.
2. The question to answer.

IMPORTANT: You must strictly adhere to the following rules:

1. Choose ONLY ONE action from the list below for each response, DO NOT perform more than one action per step.
2. Follow the exact syntax format for the selected action, DO NOT create or use any actions other than those listed.
3. ****Don't do duplicate search.**** Pay attention to the history search results.

Valid actions:

1. `<search> query </search>`: search the web for information if you consider you lack some knowledge.
2. `<answer> answer </answer>`: output the final answer if you consider you are able to answer the question. The answer should be short and concise. No justification is needed.
3. `<summary> important parts of the history turns </summary>`: summarize the history turns. Reflect the search queries and search results in you history turns, and keep the information you consider important for answering the question and generating your report. Still keep the tag structure, keep search queries between `<search>` and `</search>`, and keep search results between `<information>` and `</information>`. The history turn information for your subsequent turns will be updated accoring to this summary action.

Format:

You should pay attention to the format of your output. You can choose ****ONLY ONE**** of the following actions:

- If You want to search, You should put the query between `<search>` and `</search>`.
- If You want to summarize the history turns, You should put the summary between `<summary>` and `</summary>`.
- If You want to give the final answer, You should put the answer between `<answer>` and `</answer>`.

You can only use ONE action per response.

Format:

```
<think> thinking process </think>
[your action output]
```

Example:

```
<think> I need to answer the question, so I need to... </think>
<search> query </search>
```

Note: text between `<information></information>` is the search results from search engine after you perform a search action, ****DO NOT**** include any information in `<information></information>` in your search action.

Question: {question}

Question: {question}

History Turns: (empty if this is the first turn)

A.2 PROMPT FOR BEHAVIOR IDENTIFICATION

Prompt for Trajectory Analysis

[Instruction]

You are tasked with analyzing multi-step trajectories of a search agent's two attempts for answering the same question using search tools. One of the attempts correctly answers the question, and another attempt does not. Based on the content, please provide a detailed explanation of why one attempt succeeds and the other fails.

There are two parts in each step of the trajectory:

1. Agent output: The agent's output in this step, consists of it's thinking process and the final action.
2. Environment feedback: The feedback from the environment, including the search results wrapped in `<information>` and `</information>` tags when the agent performs a search action in this step.

The agent could perform one of the following actions in each step:

1. `<search>` query `</search>`: search the web for information
2. `<answer>` answer `</answer>`: output the final answer
3. `<summary>` important parts of the history turns `</summary>`: summarize the history turns to keep valuable information for solving the question.

Please analyze the agent's behavior in each step and provide a detailed explanation of why one attempt succeeds and the other fails.

[Question]
{question}

[Trajectory 1]
{trajectory_1}

[Evaluation Results 1]
{evaluation_results_1}

[Trajectory 2]
{trajectory_2}

[Evaluation Results 2]
{evaluation_results_2}

[Your Explanation]

Prompt for Key Reasoning Behavior Extraction

You are an expert in analyzing the behavior of a search agent. You will be provided with an explanation about a search agent's two attempts to answer the same question using search tools. The first attempt correctly answers the question, while the second attempt fails.

Based on the explanation of why trajectory 1 succeeds while trajectory 2 fails, extract the key reasoning behaviors statements implied by the explanation that lead to the success of trajectory 1. These should be clear, objective, and unambiguously verifiable.

Return the list as a JSON array of strings. Do not include markdown code fences. If there are no rule-like statements, return an empty JSON array.

[Reasoning]
{reasoning_text}

Prompt for Behavior Summarization

You are an expert in analyzing the behavior of a search agent. You are provided with a set of behaviors describing the the reasoning process and actions of the agent.

Below is a list of behaviors regarding the behavior of the search agent. Some behaviors may be duplicates or express very similar meanings. Please merge them by removing duplicates and consolidating similar behaviors, while keeping only the most essential information. When merging, discard narrow or overly specific restrictions, and retain only general behaviors that are broadly applicable.

The final rules should be clear, objective, and unambiguous, so they can be reliably used to evaluate the agent's reasoning and interaction trajectory.

Return the merged list as a JSON array of strings. Do not include markdown code fences.

```
[Behaviors]
{behaviors_text}
```

A.3 PROMPT FOR BEHAVIOR ANALYSIS FRAMEWORK

Prompt For Behavior Frequency Analysis

[Instruction]

You are tasked with analyzing a multi-step trajectory of a search agent's attempt for answering a question using search tools.

The agent can perform one of the following actions in each step:

1. `<search>` query `</search>`: search the web for information
2. `<answer>` answer `</answer>`: output the final answer
3. `<summary>` important parts of the history turns `</summary>`: summarize the history turns to keep valuable information for solving the question.

There are two parts in each step of the trajectory:

1. Agent output: The agent's output in this step, consists of it's thinking process and the final action.
2. Environment feedback: The feedback from the environment, including the search results wrapped in `<information>` and `</information>` tags when the agent performs a search action in this step.

Please act as an judge to evaluate whether the agent's thinking process and actions in this trajectory demonstrated any of following behaviors:

****behavior1: Information Verification****

The agent validates information across multiple reliable sources to ensure its conclusions are well-founded.

* ****Cross-Referencing:**** Actively seeking out and comparing multiple sources to confirm critical facts, or performing additional searches to verify the information.

* ****Citing Evidence:**** Explicitly basing its reasoning and conclusions on the information found, rather than making unsupported claims.

****behavior2: Authority Evaluation****

The agent assesses the reliability of its sources and resolves conflicting information.

* ****Detecting Conflicts:**** Identifying when different sources provide conflicting information and attempting to resolve the discrepancy.

* ****Prioritizing Authority:**** Giving more weight to official documentation, academic papers, and reputable news outlets over forums, blogs, or less reliable sources.

```

864
865
866 **behavior3: Adaptive Search**
867 The agent intelligently modifies its search strategy based on the information and
868 challenges encountered in previous steps.
869 * **Narrowing Focus:** Using initial broad search results to identify more
870 specific and effective keywords for subsequent searches.
871 * **Broadening Scope:** Widening the search terms or approach when initial
872 queries are too narrow and yield no useful results.
873
874 **behavior4: Error Recovery**
875 The agent recognizes previous errors and takes actions to correct its course.
876 * **Acknowledging Failure:** Explicitly noting when a search query or an entire
877 strategy is not yielding useful information, or some mistakes are made.
878 * **Strategic Pivoting:** Decisively abandoning a failed approach and formulating
879 a new plan to achieve the user's goal, or taking actions to correct the mistakes.
880
881 Be as objective as possible when evaluating the behaviors and do not evaluate
882 other characteristics of the response. If the behavior is not applicable for this
883 task, treat it as if the behavior is not demonstrated.
884
885 You must provide your answer with the following json format without markdown code
886 fences:
887
888 {{
889   "behavior1": "<'Yes' or 'No'>",
890   "behavior2": "<'Yes' or 'No'>",
891   "behavior3": "<'Yes' or 'No'>",
892   "behavior4": "<'Yes' or 'No'>",
893   ...
894 }}
895
896 [Question]
897 {question}
898
899 [Trajectory]
900 {trajectory}
901
902 [Your Answer]

```

A.4 PROMPT FOR LLM-JUDGE EVALUATION

Prompt For LLM-Judge Evaluation

```

903 Please determine if the predicted answer is SEMANTICALLY equivalent to the
904 labeled answer.
905 Question: {question}
906 Labeled Answer: {gt_answer}
907 Predicted Answer: {pred_answer}
908
909 {{
910   "rationale": "your rationale for the judgement, as a text",
911   "judgement": "your judgement result, can only be 'correct' or 'incorrect'"
912 }}

```

B CASE STUDY

Here is a case study for how the behavior-primed model (Qwen3 1.7B + Behavior Prime + RL) successfully solves a question, but the base model fails (Qwen3 1.7B + RL). The question is from the GAIA dataset. The trajectory for both models is provided below.

918 **Question** “What was the actual enrollment count of the clinical trial on H. pylori in acne vulgaris patients from
919 Jan–May 2018 as listed on the NIH website?”
920

921 B.1 SUCCESSFUL TRAJECTORY

923 Qwen3 1.7B + Behavior Prime + RL

925 **Step 1:**

926 **<think>**

927 Okay, so the user's looking for the enrollment count of a specific clinical trial
928 on
929 H. pylori and acne vulgaris, but they're zeroing in on the period from January to
930 May of 2018,
931 and they're specifically mentioning the NIH website. ...

932 **[Behavior 1: Information Verification]**

933 **My task is clear: verify if the quoted text exactly matches Greetham's**
934 **article. Accuracy is paramount; I'll use 'uncoupled', 'authors',**
935 **'mis-transmission', and 'veil' to zero in on the relevant section.**

936 **</think>**

937 **<search>**

938 clinical trial H. pylori acne vulgaris enrollment Jan-May 2018 NIH website

939 **<search>**

940 <information> retrieved documents </information>

941 **Step 2:**

942 **<think>**

943 I've got a few hits, though. Result #2 ... looks promising because it's linked to
944 ClinicalTrials.gov,
945 which is where I'd expect to find this kind of information. ...

946 **[Behavior 2: Authority Evaluation]**

947 **I'm aiming for the USGS's own reports or databases, like the**
948 **'Nonindigenous Aquatic Species' page, to get the most reliable data.**

949 **</think>**

950 **<search>**

951 NCT03411733 enrollment count Jan-May 2018 NIH

952 **</search>**

953 <information> retrieved documents </information>

954 **Step 3:**

955 **<think>**

956 The problem is that the search engine is not returning the number directly.
957 It's likely that the information is not readily available in the snippets.
958 So, I need to try to get to the page for NCT03411733 ...

959 **[Behavior 3: Adaptive Search]**

960 **The search engine may not have indexed the quote perfectly, or the user's**
961 **quote may differ slightly. I'll refine my strategy.**

962 **</think>**

963 **<search>**

964 NCT03411733 trial start date and recruitment numbers Jan-May 2018 NIH

965 **</search>**

966 <information> retrieved documents </information>

967 **Step 4:**

968 **<think>**

969 The last query didn't bring me directly to the ClinicalTrials.gov record.
970 I need to correct my approach and focus on the trial ID itself.
971

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

[Behavior 4: Error Recovery]

My initial search didn't deliver the goods... I need to get more precise.

</think>

<search>

NCT03411733 actual enrollment count Jan-May 2018 NIH

</search>

<information> retrieved documents </information>

Step 5:

<think>

...

</think>

<answer>

The actual enrollment count is 90 patients.

</answer>

B.2 UNSUCCESSFUL TRAJECTORY

Qwen3 1.7B + RL

Turn 1**<think>**

Okay, let's tackle this question. The user is asking for the actual enrollment count of a clinical trial on H. pylori in acne vulgaris patients from January to May 2018, specifically as listed on the NIH website.

Since the history is empty, I should start by searching for the clinical trial. The valid action here is to perform a search. The query would be something like "H. pylori clinical trial acne vulgaris enrollment Jan-May 2018 NIH".

I'll use the <search> action with that query. The search results might show the trial's details, including the enrollment numbers. I need to make sure the NIH website has this information.

</think>**<search>**

H. pylori clinical trial acne vulgaris enrollment Jan-May 2018 NIH

</search>

<information> ... Found NCT03411733 and related links ... </information>

No attempt to cross-check, only one generic search.

Turn 2**<think>**

[No reasoning; model attempts an invalid summary action.]

</think>**<summary>**

<search> H. pylori clinical trial acne vulgaris enrollment Jan-May 2018 NIH

</search>

<information>[All 10 results copied verbatim]</information>

</summary>

Model generated an invalid action format. Instead of correcting, it collapsed history incorrectly.

Turn 3**<think>**

[No new reasoning. Model repeats identical search query from Turn 1.]

</think>**<search>**

H. pylori clinical trial acne vulgaris enrollment Jan-May 2018 NIH

</search>

<information> ... Same results again ... </information>

Query was repeated with no refinement.

Turn 4**<think>**

Prematurely concludes the answer without verifying enrollment numbers.

</think>**<answer>**

The NIH clinical trial on H. pylori and acne vulgaris from January to May 2018 (NCT03411733) had an enrollment count of 100 participants.

</answer>

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

Failed - extracted "100" without checking against the authoritative NIH record. NIH/ClinicalTrials.gov was named, but not validated.

C THE USE OF LLMs

We utilized LLMs to enhance the clarity, grammar, and readability of this manuscript. The research ideas, content, and scientific contributions are entirely the work of the human authors, who reviewed and edited all LLM suggestions and retain full responsibility for the final text.