Beneficial Reasoning Behaviors in Agentic Search and Effective Training Methods to Obtain Them

Anonymous ACL submission

Abstract

Agentic search requires large language models (LLMs) to perform multi-step search to solve complex information-seeking tasks, imposing unique challenges on their reasoning capabilities. However, what constitutes effective reasoning for agentic search and how it can be learned remains unclear. In this work, we first investigate the reasoning behaviors that enable success in agentic search. By comparing successful and failed trajectories via an LLM-based analysis pipeline, we identify four beneficial behaviors: Information Verification, Authority Evaluation, Adaptive Search, and Error Recovery. Building on this, we propose Behavior Priming, a training approach that equips agentic search models with these reasoning behaviors before reinforcement learning (RL). Specifically, it collects trajectories with the identified behaviors for supervised fine-tuning (SFT), and then applies standard RL to further improve task performance. Experiments on Qwen3-1.7B and Llama3.2-3B-Instruct show that Behavior Priming yields relative improvements over direct RL by 37.2% on three web benchmarks and 6.2% on seven multi-hop QA benchmarks, and outperforms the SFT-then-RL baseline using outcome-correct trajectories for fine-tuning. Crucially, we show that these reasoning behaviors matter more than outcome correctness in the priming stage prior to RL. Further analysis reveals that Behavior Priming enhances exploration (pass@8) and test-time scaling (search step number), providing a robust foundation for RL.

1 Introduction

Agentic search (Jin et al., 2025a; Zheng et al., 2025; Li et al., 2025a) represents a new search paradigm in which large language models (LLMs) perform multi-step search actions to solve complex information needs. This process requires decomposing tasks, performing multi-step searches, and synthesizing search results into answers. Commercial systems such as ChatGPT's Deep Research and Google Search's AI Mode have rapidly gained adoption (Zhou and Li, 2024; Business Insider, 2025; Verge, 2025), and the open-source community has also made notable progress, especially in applying reinforcement learning (RL) to train agentic search models (Jin et al., 2025b; Zheng et al., 2025).

The core enabler of agentic search is the reasoning capability of LLMs. While prior works in mathematics and coding have shown that RL is highly effective at incentivizing reasoning (DeepSeek-AI et al., 2025; Yang et al., 2025), recent studies suggest that the success of RL depends heavily on whether the base models already exhibit specific reasoning patterns (Yeo et al., 2025; Gandhi et al., 2025; Wang et al., 2025b), such as verification and backtracking in math. However, agentic search presents unique challenges by requiring multi-step interaction with dynamic environments. Models need to detect useful information among abundant search results, resolve contradictions between conflicting sources, and maintain focus on task goals across extended trajectories. Despite the rapid progress in applying RL to agentic search (Jin et al., 2025b; Zheng et al., 2025; Song et al., 2025), it remains uncertain which specific reasoning behaviors are beneficial for this domain and how to systematically cultivate them in agentic models.

In this paper, we first design an LLM-based pipeline to study effective reasoning behaviors in agentic search. We employ a reasoning LLM to compare successful and failed agentic search trajectories, and extract behaviors based on these analyses. Through this process, we identify four beneficial reasoning behaviors: *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). Evaluations across diverse models and benchmarks show that these behaviors generalize consistently: models that exhibit them more frequently achieve stronger agentic performance, indicating that their importance is not tied to a specific model or dataset.

We then propose Behavior Priming, a two-stage training method that first cultivates these reasoning behaviors in agentic search models via SFT, and subsequently improves performance through RL. Specifically, we construct a large corpus of agentic search trajectories, filter for those that exhibit all four target reasoning behaviors, and fine-tune models on the resulting data. Then we apply RL on the behavior-primed model, allowing it to refine and internalize these behaviors through its own interaction experience, ultimately translating them into improved task performance.

We demonstrate the effectiveness of Behavior Priming through experiments on Qwen3-1.7B and Llama3.2-3B-Instruct across three web benchmarks and seven multi-hop QA benchmarks. We compare our method against direct RL and two SFT-then-RL baselines, including SFT on correct-outcome trajectories and direct distillation. Results demonstrate that Behavior Priming yields relative improvements over direct RL by 37.2% on web benchmarks and 6.2% on multi-hop QA benchmarks, and consistently outperforms two SFT-then-RL baselines.

Crucially, we conducted an ablation study to disentangle the impact of reasoning behaviors versus outcome correctness, revealing that behavior-priming is effective even when the trajectories containing reasoning behaviors lead to incorrect outcomes. We compare models fine-tuned on trajectories that exhibit desirable behaviors but lead to opposing outcomes: correct versus incorrect answers. Remarkably, the model behavior-primed on incorrect trajectories achieves performance comparable to its counterpart behavior-primed on correct ones. This surprising result highlights that reasoning behaviors, rather than outcome correctness, are the critical factor for unlocking RL potential.

Our analysis of training dynamics further clarifies the mechanisms behind Behavior Priming. During SFT, the frequency of beneficial reasoning behaviors significantly increases, accompanied by a concurrent rise in pass@k accuracy and step numbers. This provides a robust foundation for exploration and test-time scaling for subsequent RL. Throughout the RL phase, behavior-primed models maintain higher policy entropy, whereas non-primed models suffer from a steep decline in entropy and premature convergence. Models without priming fail to develop these reasoning behaviors endogenously during RL, highlighting the necessity of an explicit behavioral prior.

Our key contributions are as follows:

1. We identify four beneficial reasoning behaviors for agentic search with an LLM-based pipeline that compares trajectories from multiple models.

2. We propose Behavior Priming, a technique that first instills beneficial reasoning behaviors via SFT and then boosts model performance with RL.

3. We empirically demonstrate that Behavior Priming significantly unlocks a model's potential in RL, enabling higher performance by establishing a robust foundation for exploration and test-time scaling capabilities.

2 Related Work

Agentic search is an emerging search paradigm where LLMs iteratively use search-related tools to gather external information for solving complex, information-seeking tasks (Xu and Peng, 2025; OpenAI, 2024; Anthropic, 2025). These systems can be categorized into two approaches. The first is multi-agent systems (Alzubi et al., 2025; GPTResearcher, 2025; Li et al., 2025c) with a meticulously pre-defined workflow. The second approach focuses on single-agent, end-to-end systems (Zheng et al., 2025; Jin et al., 2025a; Nguyen et al., 2025) where a single LLM iteratively invokes search-related tools based history step context. Research on agentic search training has predominantly concentrated on the latter one as it's more easy and effective for end-to-end training.

To train agentic search models, prior work relies on SFT with synthesized trajectories, where strong LLMs generate reasoning and search traces that are distilled into smaller models (Asai et al., 2023; Wang et al., 2025a). More recent studies explore RL methods: some apply offline methods such as DPO in multi-agent systems (Li et al., 2025c), and most approaches train single-agent models end-to-end with online RL (Zheng et al., 2025; Jin et al., 2025b). To improve training efficiency, several works further adopt a two-stage SFT-then-RL paradigm, using SFT to initialize fundamental

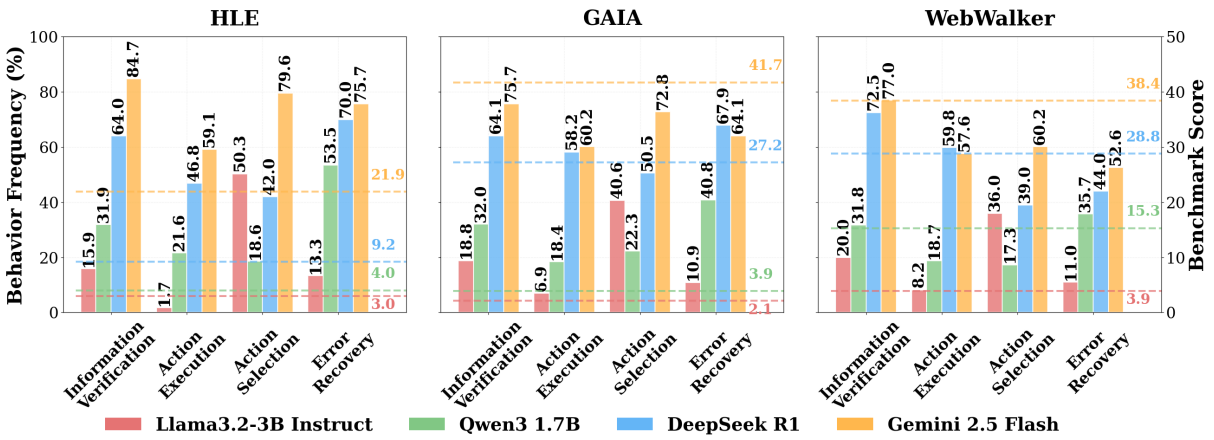

Figure 1: Comparison of different LLMs as the underlying agentic search model of our agent framework across three web agent benchmarks. For each benchmark, bars indicate the frequency of four behaviors observed in trajectories (left axis), while dashed horizontal lines indicate the benchmark score (right axis).

tool-use and reasoning capabilities before RL optimization (Li et al., 2025a; Tan et al., 2025).

From a broader perspective on LLM reasoning, RL has been shown to substantially improve performance in domains such as mathematics and coding (DeepSeek-AI et al., 2025). Recent works have shown that the effectiveness of RL is closely tied to a models initial reasoning properties: models that benefit significantly from RL usually already exhibit certain behaviors like verification and backtracking before RL (Gandhi et al., 2025; Liu et al., 2025; Yeo et al., 2025). However, relevant research has been predominantly concentrated on mathematics. For agentic domains, the beneficial reasoning behaviors that enable effective RL remain underexplored. We study this question in the context of agentic search.

3 Identifying Beneficial Behaviors in Agentic Search

To investigate what constitutes effective reasoning in agentic search, we first develop a standard agentic search framework as a foundation for subsequent studies (Section 3.1), and then design a trajectory analysis pipeline to identify beneficial reasoning behaviors (Section 3.2).

3.1 Standard Agenic Search Framework

To enable a unified investigation across diverse LLMs, we introduce a standard agentic search framework that follows common practice (Jin et al., 2025b; Song et al., 2025; Zheng et al., 2025). The framework operates iteratively. At step k, given the history context ctx_k, the model produces an output $y_k = \langle t_k, a_k \rangle$, where t_k is the reasoning process and a_k is the selected action. Actions are chosen from `search` and `answer`, with an additional `summary` action for context management. Specifically, `search` retrieves external information $info_k$, `answer` terminates the interaction, and `summary` compresses the accumulated history to manage context length, enabling support for models with limited context capacity. The complete prompts are provided in Appendix B.1.

Formally, the history context is updated as:

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

3.2 Identify Beneficial Reasoning Behaviors

To identify reasoning behaviors associated with strong agentic search capabilities, we develop an LLM-based analysis pipeline to uncover frequently exhibited patterns that distinguish successful trajectories from failed ones.

We first collect trajectory pairs with differing capability levels by integrating Gemini 2.5 Flash (as a strong model) and Qwen3-1.7B (as a weak model) into our agentic search framework. Both models are evaluated on the web agent SFT dataset of (Li et al., 2025b), which is the same dataset later used for trajectory corpus generation (Section 4.1). We then randomly select 500 questions for which Gemini 2.5 Flash succeeds while Qwen3-1.7B fails, forming paired successful (with correct final answers) and failed (with incorrect final answers) trajectories.

Next, we use reasoning LLMs to compare these trajectory pairs and extract common reasoning be-

Table 1: Examples of four beneficial behaviors in the reasoning snippets of Gemini-2.5-Flash's trajectories.

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."*

haviors associated with successful outcomes. Inspired by prior work on automated rule generation (Wang and Xiong, 2025), we employ a three-stage pipeline: (1) Trajectory comparison: we compare behaviors in successful and failed trajectories to give a detailed analysis on why one attempt succeeds while the other fails; (2) Behavior extraction: we extract key reasoning behaviors contributing to success based on this analysis; (3) Behavior consolidation: we consolidate the extracted behaviors across all trajectories by removing duplicates, merging similar behaviors, and retaining broadly applicable ones. Prompts are provided in Appendix B.2. We used Gemini 2.5 Flash for the first two stages and Gemini 2.5 Pro for the final consolidation stage. A subsequent manual review confirms that the identified behaviors recur frequently across successful trajectories.

Through this process, we identify four reasoning behaviors, with examples provided in Table 1:

1. *Information Verification*, validating search results across multiple sources, and citing evidence in the reasoning process;

2. *Authority Evaluation*, identifying conflicts among different search results and analyzing source credibility to prioritize authoritative information;

3. *Adaptive Search*, dynamically modifying search strategies based on previous search results;

4. *Error Recovery*, recognizing and correcting mistakes made in prior steps.

To assess the generality of these behaviors, we measure their frequency across multiple models on three downstream web agent benchmarks. Behavior frequency is defined as the proportion of trajectories that exhibit a given reasoning behavior. The prompt used for this measurement is provided in Appendix B.3. We evaluate Gemini 2.5 Flash, DeepSeek-R1, Llama3.2-3B-Instruct, and Qwen3-1.7B. As shown in Figure 1, we observe a strong correlation between model performance and behavior frequency. The consistency of this trend across multiple models and heterogeneous downstream datasets provides compelling evidence that these reasoning behaviors are tightly coupled with successful agentic search performance.

4 Behavior Priming: Unlocking the RL Potential for Agentic Search

Having identified the beneficial reasoning behaviors, we now investigate whether we can cultivate these behaviors to improve performance. We propose "Behavior Priming", a method that first cultivates target behaviors and then further improves performance via RL.

4.1 Instilling Behaviors via SFT

To cultivate targeted behaviors in agentic models, we collect a trajectory corpus from a strong model, filter for trajectories that exhibit all identified reasoning behaviors, and use them for SFT.

We first generate a trajectory corpus on two types of tasks: web agent tasks and multi-hop QA tasks. Questions and ground-truth answers for both datasets are drawn from the SFT dataset of (Li et al., 2025b). For each question, we sample 10 trajectories using Gemini 2.5 Flash.

We then evaluate each trajectory for the presence of each behavior and select for those that exhibit all four behaviors to construct the behavior-priming dataset. For evaluation, we concatenate the model outputs and search results at each step to form a complete trajectory, and prompt an LLM to assess whether the reasoning process at any step demonstrates each target behavior. Prompt is provided in Appendix B.3. We select trajectories that

exhibit all behaviors as training data.

Following data preparation, we train models using supervised fine-tuning. We treat each step in a trajectory as an independent training sample: for a trajectory $\mathcal{T}_i = (\langle x_1^i, y_1^i \rangle, \langle x_2^i, y_2^i \rangle, \ldots, \langle x_{L_i}^i, y_{L_i}^i \rangle)$ where x_i, y_i are the input and output, we aggregate all input-output pairs to construct the SFT dataset:

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

4.2 Reinforcement Learning

Building on the cultivated reasoning behaviors, we use RL to further enhance the model's performance by allowing it to learn from its own exploration experience.

We train models with the GRPO (Shao et al., 2024) algorithm. For each question q, we rollout a group of G trajectories $\{\mathcal{T}_i\}_{i=1}^G$. The policy is updated by aggregating over all steps from the rollout trajectories. The loss function is defined as:

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

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

We employ an outcome-based reward. An LLM-judge evaluates the final answer a_i against ground-truth, assigning a binary reward R_i of 1 for a correct answer and 0 otherwise. The reward R_i and the advantage $\hat{A}_i$ are constant for all steps in trajectory $\mathcal{T}_i$. For datasets, we use the web agent RL dataset of (Li et al., 2025b) for the web agent tasks and the dataset from (Zheng et al., 2025) for multi-hop QA tasks.

5 Experiments

In this section, we present a comprehensive evaluation of our Behavior Priming method. We compare it with direct RL and strong SFT-then-RL baselines on web agent and multi-hop QA tasks (Section 5.1), disentangle the reasoning process from outcome correctness (Section 5.2), and study the impact of SFT data scale (Section 5.3). Detailed experimental setup is in Appendix A.

5.1 Baselines

To evaluate the effect of our behavior-centric data curation method, we compare it against some widely adopted methods for agentic training.

Direct RL We include the *Direct RL* baseline that applies RL directly without any preceding supervised fine-tuning. This baseline is introduced to explicitly assess the role of the behavior cultivation stage of SFT, and to examine whether models can spontaneously acquire the beneficial reasoning behaviors through RL, or achieve strong performance without the behavioral prior.

SFT-then-RL Baselines We compare our behavior-centric method with two common SFT-then-RL baselines with different fine-tuning data selection strategies: *Distillation*, which randomly samples trajectories from the corpus generated by the stronger model; and *Outcome-driven*, which filters for trajectories with the correct final answer. Both baselines select trajectories for fine-tuning from the same Gemini 2.5 Flash-generated corpus as our method (Section 4.1), ensuring a strong and consistent point of comparison. Detailed dataset statistics are reported in Table 7.

Prior Works To evaluate our entire training framework (e.g., agent framework, datasets, RL algorithm), we compare against prior agentic search systems trained with RL, including Search-R1 (Jin et al., 2025b), R1-searcher (Song et al., 2025), and Deepresearcher (Zheng et al., 2025). We directly adopt the evaluation results reported in (Zheng et al., 2025).

5.2 Disentangling Reasoning Behaviors from Outcome Correctness

To disentangle the influence of reasoning processes from final outcomes, and to assess the synergy between high-quality reasoning and outcome correctness, we evaluate two variants of our method: *Behavior Prime (Incorrect)*, which fine-tunes models on trajectories exhibiting all four reasoning behaviors but resulting in incorrect outcomes; and *Behavior Prime (Correct)*, which fine-tunes models on trajectories exhibiting all four behaviors and resulting in correct outcomes. Both methods apply standard RL after the SFT stage.

5.3 Impact of SFT Data Size

We also investigated how the scale of the SFT data affects the efficacy of Behavior Priming. We fine-tuned Qwen3-1.7B on subsets of our Behavior Prime dataset on the web task with varying sizes: 5k, 10k, and the full 20k samples. We then compared their performance against the Direct RL baseline (denoted as 0k).

Table 2: Performance on web agent benchmarks before and after RL. The overall score is calculated as the average of seven benchmarks. The scores in the *Before RL* section are performance right after the SFT stage, and the final performance scores are under in *After RL* section.

		GAIA				WebWalkerQA	HLE	Overall
Method	**Base Model**	**Level 1**	**Level 2**	**Level 3**	**Avg.**			
Before RL (After SFT)								
Direct RL (No SFT)	Qwen3-1.7B	7.7	1.9	0.0	3.9	15.3	4.0	7.7
Distillation	Qwen3-1.7B	**12.8**	7.7	0.0	8.7	23.5	**5.4**	12.5
Outcome-driven	Qwen3-1.7B	10.3	**9.6**	0.0	8.7	**24.3**	4.8	**12.6**
Behavior Prime	Qwen3-1.7B	**12.8**	7.7	0.0	8.7	22.0	4.6	11.8
After RL								
Direct RL	Qwen3-1.7B	15.4	11.5	0.0	11.7	26.1	3.9	13.9
Distillation	Qwen3-1.7B	18.0	11.5	**16.7**	14.6	33.2	7.4	18.4
Outcome-driven	Qwen3-1.7B	23.1	17.3	0.0	17.5	36.8	5.8	20.0
Behavior Prime	Qwen3-1.7B	**28.2**	**21.2**	0.0	**21.4**	**37.2**	**7.8**	**22.3**
Direct RL	Llama3.2-3B-Instruct	12.8	11.5	8.3	11.7	24.7	7.0	14.5
Outcome-driven	Llama3.2-3B-Instruct	15.4	11.5	16.7	13.6	28.8	9.8	13.3
Distillation	Llama3.2-3B-Instruct	15.4	**21.5**	**16.7**	**18.4**	26.5	6.6	17.2
Behavior Prime	Llama3.2-3B-Instruct	**25.6**	13.4	**16.7**	**18.4**	**33.8**	**7.5**	**19.9**

Table 3: Performance on multi-hop QA benchmarks. The overall score is the average score of seven benchmarks.

Method	**Base Model**	**2wiki**	**Bamboogle**	**Hotpotqa**	**Musique**	**NQ**	**PopQA**	**TriviaQA**	**Overall**
Prior work baselines									
Search-R1-base	Qwen2.5-7B-Base	47.9	57.6	63.0	27.5	60.0	47.0	76.2	54.2
Search-R1-instruct	Qwen2.5-7B-Instruct	48.8	47.2	52.5	28.3	49.6	44.5	49.2	45.7
R1-Searcher	Qwen2.5-7B-Base	65.8	65.6	53.1	25.6	52.3	43.4	79.1	55.7
DeepResearcher	Qwen2.5-7B-Instruct	66.6	72.8	64.3	29.3	61.9	52.7	85.0	61.8
Our Work									
Direct RL	Qwen3-1.7B	66.8	64.0	61.7	25.0	72.7	53.5	**87.9**	61.7
Distillation	Qwen3-1.7B	65.6	**74.4**	64.8	26.9	**74.0**	56.1	85.7	63.9
Outcome-driven	Qwen3-1.7B	70.3	68.8	65.0	29.7	72.7	54.7	86.3	63.9
Behavior Prime	Qwen3-1.7B	**73.8**	70.4	**67.0**	**30.7**	73.6	**56.6**	86.9	**65.5**
Direct RL (No SFT)	Llama3.2-3B-Instruct	74.0	72.0	65.8	31.5	74.8	56.4	90.4	66.4
Distillation	Llama3.2-3B-Instruct	77.7	76.0	**71.3**	32.0	78.9	55.7	89.3	68.7
Outcome-driven	Llama3.2-3B-Instruct	69.9	72.0	69.9	30.5	77.3	60.0	89.5	67.0
Behavior Prime	Llama3.2-3B-Instruct	75.6	**80.0**	69.5	**38.1**	**82.0**	**60.4**	**90.6**	**70.9**

Table 4: Performance comparison between models behavior-primed on correct vs. incorrect trajectories before RL.

		GAIA				WebWalkerQA	HLE	Overall
Method	**Base Model**	**Level 1**	**Level 2**	**Level 3**	**Avg.**			
Before RL (After SFT)								
Behavior Prime (Incorrect)	Qwen3-1.7B	7.7	**11.5**	0.0	8.7	15.3	3.5	9.3
Behavior Prime (Correct)	Qwen3-1.7B	10.3	9.6	0.0	8.7	19.1	**6.2**	11.3
After RL								
Behavior Prime (Incorrect)	Qwen3-1.7B	30.8	**26.9**	8.3	**27.2**	35.5	**7.8**	23.5
Behavior Prime (Correct)	Qwen3-1.7B	**30.8**	23.1	**16.7**	25.2	**37.8**	**7.8**	**23.6**

6 Results

6.1 Main Results

In this section, we present our main experimental results, demonstrating that Behavior Priming consistently leads to best performance, and its effect holds even when priming trajectories yield incorrect outcomes. These results provide strong evidence that the identified reasoning behaviors are able to enable strong agentic search performance.

Behavior Priming Effectively Increases Headroom for RL As shown in Table 2 and Table 3, Behavior Priming consistently achieves substantially higher performance than Direct RL. In particular, the overall scores on both Llama-3.2-Instruct and Qwen3-1.7B exceed those of Direct RL by over 37% on web benchmarks and 6.2% on multi-hop QA benchmarks. Moreover, Behavior Priming consistently outperforms strong SFT-then-

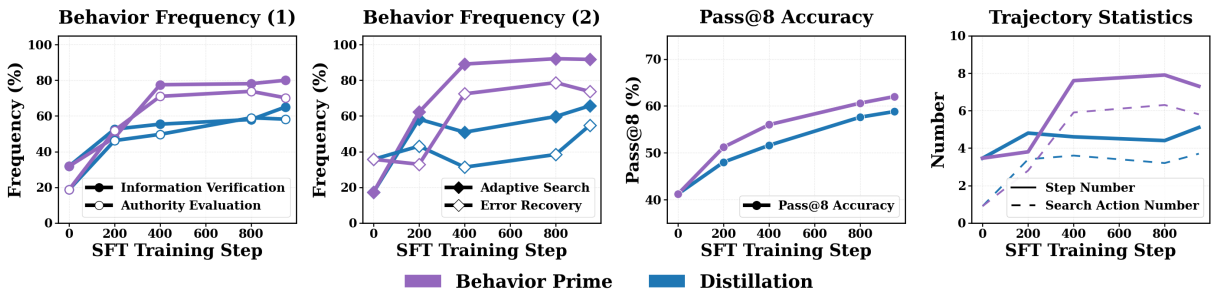

Figure 2: Qwen3 1.7B + *Distillation* and Qwen3 1.7B + *Behavior Prime*'s behavior frequencies, pass@8 accuracy, and trajectories statistics (average step number and search action number per trajectory) on the WebWalkerQA benchmark during the SFT process.

RL baselines, including *Distillation* and *Outcome-driven*. These results demonstrate that Behavior Priming establishes a robust foundation through behavior-centric SFT, enabling more effective improvement during subsequent RL training.

In addition, comparisons with prior agentic search systems show that our training framework achieves competitive overall performance, indicating it provide a solid and reliable foundation for agentic search training.

Reasoning Behavior is More Important than Outcome Correctness As shown in Table 2, while *Outcome-driven* achieves the strongest performance after SFT, this advantage does not lead to the best performance after RL. During the subsequent RL training, it is surpassed by our *Behavior Prime* method. This demonstrated that although training with answer-correct trajectories provides a powerful boost after SFT, cultivating reasoning behaviors is the key to enabling effective exploration and achieving superior performance during the subsequent RL phase.

Notably, as shown in Table 4, our reasoning behavior and outcome correctness disentanglement study demonstrates that, although the *Behavior Prime (Incorrect)* variant achieved suboptimal performance after SFT, it ultimately achieved comparable performance with the *Behavior Prime (Correct)* variant after RL. This confirms that the gains from behavior priming are driven by the reasoning patterns, independent of outcome correctness.

Scaling SFT Data Enhances Behavior Priming Figure 3 shows that increasing the size of the SFT dataset for behavior priming improves final performance, indicating that the benefits of our method scale with additional data. However, performance on WebWalkerQA plateaus beyond 5k data scale. We hypothesize that the SFT stage mainly establishes a behavioral foundation for RL, and once the targeted behaviors are learned, additional priming yields diminishing returns.

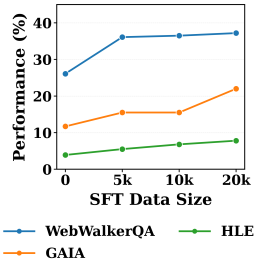

Figure 3: Qwen3-1.7B's performance when behavior primed with different SFT data sizes.

6.2 Analysis with Training Dynamics

To better understand our method, we examine how reasoning behavior frequencies, exploration abilities, trajectory length, and model performance evolve over the SFT and the RL phase.

Behavior Priming Encourages Exploration We track model evolution during SFT by monitoring behavior frequencies, Pass@8, and trajectory statistics. As shown in Figure 2, although both *Distillation* and *Behavior Prime* improve Pass@8 and trajectory step number alongside rising behavior frequencies, the gains are substantially larger for *Behavior Priming*. This indicates that explicitly priming reasoning behaviors enables the model to explore more diverse paths and allocate greater test-time compute through more search actions.

Sustained Exploration Translates to a Higher RL Ceiling We then analyzed the training dynamics during RL, monitoring the policy entropy and validation performance. Figure 4 (a) shows that the behavior-primed model starts with higher entropy and is less prone to collapse. In contrast, the non-behavior-primed model (Direct RL)'s entropy collapses rapidly. This indicates that be-

Table 5: Qwen3-1.7B's behavior frequency (averages across three benchmarks) and benchmark performance with standard Direct RL and behavior-guided RL (with additional process reward encourages four reasoning behaviors).

Model	Behavior Frequency				Benchmark 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** (+10.8)	**16.3** (+12.4)	3.9 (-0.1)
Qwen3-1.7B + behavior guided RL	67.3 (+39.0)	50.8 (+32.7)	82.8 (+64.6)	86.5 (+46.8)	15.4 (+0.1)	7.8 (+3.9)	**4.4** (+0.4)

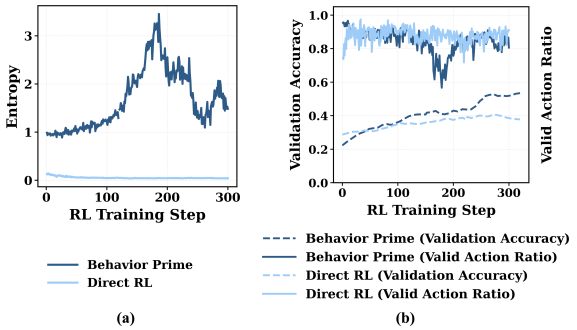

Figure 4: The entropy, validation accuracy, and valid action (percentage of correct-format actions) during RL of Qwen3-1.7B with *Behavior Prime* and *Direct RL*.

havior priming endows the model with richer exploration, allowing it to seek diverse strategies rather than prematurely converging to a suboptimal policy. This sustained exploration directly translates into performance gains, as shown in Figure 4 (b): while the base model quickly reaches a low plateau, the behavior-primed model converges more slowly to a higher performance ceiling.

Another finding is that while the non-behavior-primed model (Direct RL) starts with a lower valid action ratio (the percentage of steps with correct action format) compared to the behavior-primed model, it successfully masters the required output format within 20 steps and maintains a high valid action ratio thereafter. Conversely, the behavior-primed model even exhibits a less stable valid action ratio during RL. This indicates that the improvements from Behavior Priming arise from enhanced reasoning behaviors rather than mere familiarity with tool-use syntax.

7 Discussion: Is Process Rewards an Alternative for Behavior Guidance?

We also explore alternative ways to use these beneficial behaviors to improve model performance. Curious about whether the model could spontaneously develop these reasoning behaviors during RL, we measured the change in behavior frequencies for a non-behavior-primed model after Direct RL. Results in Table 5 show that most of the frequencies even dropped after RL. Considering that our behavior frequency evaluation is based on process-level, we investigate whether we could guide the emergence of these behaviors with an additional process reward. Specifically, we combined the standard outcome-based reward with a process-based reward that encourages the exhibition of reasoning behaviors. We measure the number of different behaviors in a trajectory N, and define the final reward as $R = R_{\text{outcome}} + 0.1 \times N$.

As shown in Table 5, this approach successfully increases the behavior frequency, but leads to worse performance compared to standard RL. This result aligns with 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 reward but fails to grasp their essence for problem-solving. This finding reveals that our Behavior Priming method is a more effective way to improve model performance.

8 Conclusion

We identify four beneficial reasoning behaviors for agentic search and propose Behavior Priming, a method that cultivates these behaviors through SFT to enable more effective RL. Experiments across multiple models and ten benchmarks show that Behavior Priming achieves the strongest performance compared to other SFT-then-RL baselines, and that cultivating reasoning behaviors is more important than teaching correct answers in preparing models for RL. Further analysis reveals that Behavior Priming establishes a robust foundation for RL by enabling richer exploration, as reflected by higher Pass@8 and more search steps that support test-time scaling. Together, these findings highlight a promising direction for targeted post-training strategies that enhance reasoning behaviors and improve the effectiveness of RL in training capable agentic models.

9 Limitation

Model Scale Our experiments focus on relatively small models (1.7B and 3B parameters) due to computational constraints. While this allows for controlled experimentation and demonstrates proof-of-concept, the extent to which our findings scale to larger models (7B+ parameters) remains an open question. Nevertheless, since the proposed method targets general reasoning behaviors rather than model-specific heuristics, we expect it to remain applicable as model capacity increases.

RL Training Scale and Hyperparameters : Due to computational resource constraints, our reinforcement learning phase was conducted for 300 steps with a group size of 8 using the GRPO algorithm. It is possible that further scaling of the training steps or larger group sizes could reveal pronounced performance gains, though current results already provide compelling evidence of our method's efficacy.

10 Ethical considerations

Agentic search models are dependent on the underlying search systems, so systemic biases or misinformation present in indexed web content may still be synthesized into final answers. Besides, the increased efficiency in agentic information seeking could potentially be misused for the discovery of sensitive information. We use only established public datasets and benchmarks, which are curated for research purposes and align with common safety and privacy standards.

We utilize three primary external artifacts in our study: the AFM-WebAgent-SFT-Dataset[1], the AFM-MHQA-Agent-SFT-Dataset[2], and the RL dataset derived from DeepResearcher[3]. All artifacts are released under the Apache License 2.0, which permits use, modification, and distribution for research purposes. Our use of these datasets–specifically for generating our SFT dataset and for RL training–is strictly consistent with their intended use of advancing research in agentic reasoning and search capabilities.

[1] https://huggingface.co/datasets/PersonalAILab/AFM-WebAgent-SFT-Dataset

[2] https://huggingface.co/datasets/PersonalAILab/AFM-MHQA-Agent-SFT-Dataset

[3] https://github.com/GAIR-NLP/DeepResearcher

References

Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, and 1 others. 2025. Open deep search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*.

Anthropic. 2025. Claude can now search the web.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Business Insider. 2025. Apple and google disagree on ai cutting into search. *Business Insider*. Accessed: 2025-09-22.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *Preprint*, arXiv:2505.10978.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*.

GPTResearcher. 2025. Say goodbye to hours of research.

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

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

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

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin,

Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, and 1 others. 2025a. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, and 1 others. 2025b. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025c. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *Preprint*, arXiv:2212.10511.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

Xuan-Phi Nguyen, Shrey Pandit, Revanth Gangi Reddy, Austin Xu, Silvio Savarese, Caiming Xiong, and Shafiq Joty. 2025. Sfr-deepresearch: Towards effective reinforcement learning for autonomously reasoning single agents. *arXiv preprint arXiv:2509.06283*.

OpenAI. 2024. Introducing chatgpt search.

Long Phan and 1 others. 2025. Humanity's Last Exam. *ArXiv*, abs/2501.14249.

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*, pages 5687–5711.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *Preprint*, arXiv:2503.05592.

Zhiwen Tan, Jiaming Huang, Qintong Wu, Hongxuan Zhang, Chenyi Zhuang, and Jinjie Gu. 2025. Rag-r1: Incentivizing the search and reasoning capabilities of llms through multi-query parallelism. *Preprint*, arXiv:2507.02962.

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*, 10:539–554.

Verge. 2025. Google searches are falling in safari for the first time ever probably because of ai. *The Verge*. Accessed: 2025-09-22.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025a. Chain-of-retrieval augmented generation. *Preprint*, arXiv:2501.14342.

Tevin Wang and Chenyan Xiong. 2025. Autorule: Reasoning chain-of-thought extracted rule-based rewards improve preference learning. *Preprint*, arXiv:2506.15651.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025b. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and 1 others. 2025a. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025b. Webwalker: Benchmarking llms in web traversal. *Preprint*, arXiv:2501.07572.

Renjun Xu and Jingwen Peng. 2025. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*.

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

Tao Zhou and Songtao Li. 2024. Understanding user switch of information seeking: From search engines to generative ai. *Journal of librarianship and information science*, page 09610006241244800.

A Detailed Experiments Setup

Benchmarks For web benchmarks, we use **WebWalkerQA** (Wu et al., 2025b), **GAIA** (Mialon et al., 2023)(following (Li et al., 2025c; Wu et al., 2025a; Li et al., 2025b) to use the 103 text-based examples), and **Humanity's Exam (HLE)** (Phan et al., 2025). For multi-hop QA benchmarks, we use **NQ** (Kwiatkowski et al., 2019), **TQ** (Joshi et al., 2017), **HotpotQA** (Yang et al., 2018), **2Wiki** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022), **Bamboogle** (Press et al., 2023), and **PopQA** (Mallen et al., 2023). We followed (Zheng et al., 2025) to use 512 examples from the development sets of NQ, TQ, HotpotQA, 2Wiki, MuSiQue, and PopQA and all 125 samples from Bamboogle.

Evaluation The max step number for the trajectory is 25 for web benchmarks and 15 for multi-hop QA benchmarks. The temperature is 0.0 for the main evaluation, and 1.0 for the pass@k evaluation. We use GPT-4o-mini to score the final answers. Prompts are provided in Appendix B.4.

Training Details We adopt Qwen3-1.7B (Yang et al., 2025) and Llama3.2-3B-Instruct (Meta, 2024) as base models. For SFT, each model was fine-tuned for 3 epochs with a batch size of 8. The RL training was conducted on the verl-agent (Feng et al., 2025) framework for 300 steps, with a batch size of 32 and a GRPO group size of 8. All RL experiments were performed on 8 NVIDIA H100 GPUs, with each training run taking approximately 20 hours to complete.

B Prompts

B.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)
```

B.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}
```

B.3 Prompt For Behavior Frequency Evaluation

Prompt For Behavior Frequency Evaluation

```
[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.

**behavior3: Adaptive Search**
The agent intelligently modifies its
search strategy based on the information
and challenges encountered in previous
steps.
* **Narrowing Focus:** Using initial broad
search results to identify more specific
and effective keywords for subsequent
searches.
* **Broadening Scope:** Widening the
search terms or approach when initial
queries are too narrow and yield no useful
results.

**behavior4: Error Recovery**
The agent recognizes previous errors and
takes actions to correct its course.
* **Acknowledging Failure:** Explicitly
noting when a search query or an entire
strategy is not yielding useful
information, or some mistakes are made.
* **Strategic Pivoting:** Decisively
abandoning a failed approach and
formulating a new plan to achieve the
user's goal, or taking actions to correct
the mistakes.

Be as objective as possible when
evaluating the behaviors and do not
evaluate other characteristics of the
response. If the behavior is not
applicable for this task, treat it as if
the behavior is not demonstrated.

You must provide your answer with the
following json format without markdown
code fences:

{
  "behavior1": "<'Yes' or 'No'>",
  "behavior2": "<'Yes' or 'No'>",
  "behavior3": "<'Yes' or 'No'>",
  "behavior4": "<'Yes' or 'No'>",
  ...
}
```

```
[Question]
{question}

[Trajectory]
{trajectory}

[Your Answer]
```

B.4 Prompt for Evaluation

Prompt For LLM-Judge Evaluation on Web Tasks

```
Please determine if the predicted answer
is SEMANTICALLY equivalent to the labeled
answer.
Question:  {question}
Labeled Answer:  {gt_answer}
Predicted Answer: {pred_answer}

{
"rationale": "your rationale for the
judgement, as a text",
"judgement": "your judgement result, can
only be 'correct' or 'incorrect'"
}
```

Prompt For LLM-Judge Evaluation on Multi-hop QA Tasks

```
You will be given a question and its
ground truth answer list where each item
can be a ground truth answer. Provided a
pred_answer, you need to judge if the
pred_answer correctly answers the question
based on the ground truth answer list.
You should first give your rationale for
the judgement, and then give your
judgement result (i.e., correct or
incorrect).

Here is the criteria for the judgement:
1. The pred_answer doesn't need to be
exactly the same as any of the ground
truth answers, but should be semantically
same for the question.
2. Each item in the ground truth answer
list can be viewed as a ground truth
answer for the question, and the
pred_answer should be semantically same to
at least one of them.

question: {question}
ground truth answers: {gt_answer}
pred_answer: {pred_answer}

The output should in the following json
format:
{
"rationale": "your rationale for the
judgement, as a text",
"judgement": "your judgement result, can
only be 'correct' or 'incorrect'
}
```

C Case Study

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

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

C.1 Successful Trajectory

Qwen3 1.7B + Behavior Prime

Step 1:
<think>

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

[Behavior 1: Information Verification]
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.
</think>

<search>

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

<search>

<information> retrieved documents
</information>

Step 2:
<think>

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

[Behavior 2: Authority Evaluation]
Im aiming for the USGSs own reports or databases, like the Nonindigenous Aquatic Species page, to get the most reliable data.
</think>
<search>

NCT03411733 enrollment count Jan-May 2018 NIH

</search>

<information> retrieved documents
</information>

Step 3:
<think>

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

[Behavior 3: Adaptive Search]
The search engine may not have indexed the quote perfectly, or the users quote may differ slightly. Ill refine my strategy.
</think>
<search>

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

</search>

<information> retrieved documents
</information>

Step 4:
<think>

The last query didnt bring me directly to the ClinicalTrials.gov record.
I need to correct my approach and focus on the trial ID itself.

[Behavior 4: Error Recovery]
My initial search didnt 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>
Result: Correct.

C.2 Unsuccessful Trajectory

Qwen3 1.7B + Direct 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>
Result: Incorrect. Extracted "100" without checking against the authoritative NIH record. NIH/ClinicalTrials.gov was named, but not validated.

D Contrasting Single-Behavior Priming with Composite-Behavior Priming

We also investigated whether the composite set of four behaviors was truly necessary, or if the benefit could be achieved by priming with a single behavior. We identified Information Verification (IV) as the most common behavior in our trajectory corpus and selected it for this contrastive study. We used two SFT datasets on the web task of equal size (10k samples each): 1) *Behavior Prime (IV-Only)-10k*: Trajectories selected only for exhibiting Information Verification. 2) *Behavior Prime-10k*: The 10k subset dataset in Section 5.3 that

Table 6: Performance after RL on web agent benchmarks for priming with single behavior and all behaviors. The Behavior Prime (IV-Only)-10k is primed on the Information Verification behavior only with 10k data samples.

Method	**Base Model**	**GAIA**	**WebWalkerQA**	**HLE**	**Overall**
Direct RL	Qwen3-1.7B	11.7	26.1	3.9	13.9
Behavior Prime (IV-Only)-10k	Qwen3-1.7B	14.7	30.7	6.8	17.4
Behavior Prime-10k	Qwen3-1.7B	**15.5**	**36.5**	**7.2**	**19.7**

Table 7: Behavior frequencies, outcome accuracy, and trajectory statistics for the SFT datasets on both tasks. The number of steps is the number of data samples used for training (#Steps = #Traj. * Avg. Steps).

Web Agent Task	Info. Verif.	Auth. Eval.	Adapt. Search	Error Rec.	Accuracy	Avg. Steps	#Traj.	#Steps
Distillation	71.7	42.2	52.3	36.2	40.0%	4.6	4.3k	20k
Outcome-driven	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
Multi-hop QA Task								
Distillation	93.0	29.8	25.6	4.3	58.1%	2.5	4.1k	10k
Outcome-driven	94.9	25.8	25.3	2.6	100.0%	2.5	4.1k	10k
Behavior Prime	100.0	100.0	100.0	100.0	61.8%	4.6	2.2k	10k

contains all four behaviors.

We then applied the SFT + RL pipeline to Qwen3-1.7B using both datasets. Results in Table 6 show that while priming with only the most frequent behavior (*Behavior Prime (IV-Only)-10k*) enhances performance compared to the direct RL baseline, it is consistently outperformed priming with all four behaviors (*Behavior Prime-10k*). This confirms that the superior performance of our method is unlocked by instilling the complete, synergistic set of all behaviors, which are critical for building a robust foundation for RL.

E Use of AI Assistants

We used AI tools in paper writing for language refinement to improve grammatical accuracy and phrasing. All core methodologies, experimental designs, and data interpretations were developed solely by the human authors.