Beyond Outcome Rewards: Step-Level Self-Distilled Policy Optimization for Deep Search Agents

Haoze Wu[1] **Chuqiao Kuang**[2] **Tianyi Zhuang**[2] **Xiaoguang Li**[2]

[1]The Hong Kong University of Science and Technology [2]Huawei Technologies Ltd.

Abstract

Deep search agents operate over trajectories spanning dozens of steps, yet standard reinforcement learning provides only a single outcome reward per trajectory — supervision that is far too sparse for effective credit assignment. On-policy self-distillation (OPSD) addresses this by using the model's own logits as dense token-level teachers, but extending it to search agents introduces a fundamental tension: the teacher, having access to privileged information such as the correct answer, produces a distribution that differs systematically from the student's exploration-based reasoning, and naive distillation causes the student to inherit this information asymmetry rather than learn better search strategies. We resolve this tension through two contributions. First, we construct Evidence Anchors — concise, step-level evidence snippets extracted from the web — as privileged information that captures key reasoning steps without revealing the entire answer path. Second, we propose Step-Level Self-Distilled Policy Optimization (SSPO), which converts teacher–student disagreement into step-level advantage weights within GRPO, applied exclusively to incorrect trajectories. This design decouples what to update from how much to update: the outcome reward determines the direction of policy change, while the teacher modulates its magnitude at each step. Correct trajectories are left untouched, preserving their diversity. On Qwen3-8B, SSPO consistently outperforms GRPO across BrowseComp, GAIA, and FRAMES, surpassing or matching GRPO trained with twice as many gradient steps while adding only about 5% overhead per step from a single additional forward pass.[1]

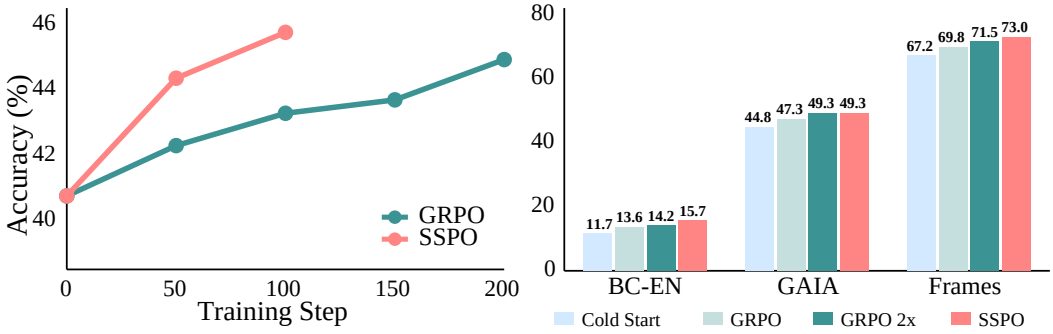


Figure 1: **Comparison of performance across training methods. Left:** After introducing step-level self-distilled advantage weights (SSPO), performance improves substantially faster during training, surpassing GRPO trained for 200 steps after only 100 steps. **Right:** We report results for the cold-start model, GRPO after 100 and 200 steps, and SSPO after 100 steps.

[1]Code is available at `https://github.com/hkust-nlp/SSPO`. Correspondence to: `hwuds@connect.ust.hk`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Recent Large Language Model paradigms have rapidly evolved from generating appropriate responses to solving real-world tasks through tool use across domains such as coding, office productivity, finance, and research [18, 33, 41, 4, 2, 7, 34, 30, 8]. Among these applications, web search agents must answer vague, underspecified queries by actively exploring the open web — issuing searches, inspecting pages, and refining their reasoning over trajectories that often span dozens of steps [43]. Training agents to perform such long-horizon search reliably requires post-training beyond Supervised Fine-Tuning (SFT), making reinforcement learning (RL) the dominant approach [26, 20, 19, 40]. However, RL in this setting faces a fundamental challenge: trajectories containing 20+ steps receive only a single binary reward. With supervision this sparse, the model receives little guidance on which steps led to success or failure, making credit assignment the central bottleneck for deep search agents.

A natural way to address sparse rewards is to introduce denser supervision. SFT provides token-level feedback through teacher trajectories, but its Off-Policy nature introduces exposure bias and limits generalization to unseen queries [36, 6, 48, 38]. On-Policy Distillation (OPD) [27, 39, 23] combines both benefits: the student performs on-policy rollouts while a teacher model provides token-level scoring. More recent work extends this to self-distillation, where the teacher is constructed from the student itself — using privileged information such as reference solutions or environment feedback as a prefix [63, 54, 13, 38, 35, 17], eliminating the need for a separate teacher model.

Despite this progress, existing self-distillation methods have been developed and evaluated primarily on single-turn reasoning tasks such as math and code. Two properties make self-distillation feasible in those settings: the gap between teacher and student behavior is modest, and a natural form of privileged information — reference solutions or execution feedback — already exists. Extending self-distillation to multi-turn search agents breaks both assumptions. The information asymmetry becomes extreme: the teacher, seeing curated evidence and the answer, collapses to 3.5 tool calls per trajectory, while the student, navigating the open web, requires 17.7 (Table 1). Directly distilling this collapsed distribution causes the student to mimic the teacher's brevity; when privileged information is unavailable, it abandons tool use prematurely rather than learning better search strategies. An equally serious problem compounds the first: unlike math or code, open-ended web search has no ready-made privileged information. A form of supervision that is both informative to the teacher and safe to distill to the student does not naturally exist, and constructing one is itself an open challenge.

Our key insight is that search supervision is naturally defined at the level of information-seeking actions rather than token generation. We therefore align privileged supervision with the structure of search through two design choices. First, we construct **Evidence Anchors**: compact, step-level evidence snippets extracted from the web that capture information needed to answer a question without revealing the answer path. Each question is associated with a small set of such anchors, serving as prefixes for the self-teacher. Second, we propose **Step-Level Self-Distilled Policy Optimization** (SSPO). Rather than minimizing divergence between teacher and student distributions, SSPO converts teacher–student disagreement into step-level advantage weights within GRPO [37, 11], applied only to incorrect trajectories. This design decouples update direction from update magnitude: the outcome reward determines whether to reinforce or suppress a trajectory, while the teacher modulates how much each step contributes. Correct trajectories are left untouched, preserving behavioral diversity.

We evaluate SSPO on Qwen3-8B [53] across three information-seeking benchmarks — BrowseComp [43], GAIA [29], and FRAMES [16]. SSPO consistently outperforms GRPO across all benchmarks, and notably surpasses GRPO trained with twice as many gradient steps while adding only about 5% computational overhead per training step. Our ablation studies further validate two key design decisions. First, directly matching the teacher distribution collapses the student's tool use and underperforms even GRPO, confirming that decoupling update magnitude from direction is essential. Second, step-level advantage weights consistently outperform token-level counterparts, whose fine-grained signals are misaligned with the natural unit of information-seeking actions.

In summary, our contributions are: **(a)** We show that extending self-distillation to multi-turn search breaks the assumptions underlying single-turn self-distillation, requiring a fundamentally different design. **(b)** We introduce Evidence Anchors: compact, step-level evidence snippets as privileged information for open-ended web search, aligned with the natural unit of information-seeking actions. **(c)** We propose SSPO, which uses step-level self-distilled signals as advantage weights rather than optimization targets, achieving superior sample efficiency over GRPO.

2 Preliminaries

2.1 Search Agent

We build our search agent based on the ReAct paradigm [56, 50], where the model interleaves reasoning and actions until producing the final answer. Our action space includes two tools connected to the real web: `search` and `browse`, their details could be found in Appendix C. After receiving a user query, the agent attempts to provide an accurate answer through multiple *Thought Action Observation* steps. An agent with T steps can be formulated as follows:

$$H_T = \{\mathtt{q}_{user}, \mathtt{t}_1, \mathtt{a}_1, \mathtt{o}_1, \ldots, \mathtt{t}_{T-1}, \mathtt{a}_{T-1}, \mathtt{o}_{T-1}, \mathtt{t}_T\}$$

At time step τ, the agent generates a thought $\mathtt{t}_\tau$ and a tool call $\mathtt{a}_\tau$ (if it is not the final step) conditioned on the full interaction history, i.e., $\mathtt{t}_\tau, \mathtt{a}_\tau \sim \pi(\mathtt{t}, \mathtt{a} \mid H_{\tau-1})$. The output of the tool is treated as the observation $\mathtt{o}_\tau$ for this step.

2.2 On-Policy Learning

Agentic Reinforcement Learning. Deep search, similar to math and logic tasks [37, 50], is typically formulated under the Reinforcement Learning with Verifiable Rewards (RLVR) setting. In this setting, a correctness reward can be assigned by evaluating whether the agent's prediction is semantically consistent with the ground truth: $R_{\text{correct}} = \texttt{is_equal(pred, ground_truth)} \in \{0, 1\}$. In addition, training often incorporates a format reward R_{format} to encourage adherence to the ReAct paradigm [56]. The final training reward is defined as:

$$R_{final} = R_{correct} + 0.2 \times R_{format} \tag{1}$$

Similar to other works [26, 20, 19], we use the Group Relative Policy Optimization (GRPO) algorithm to update the model [37, 11]. It addresses the challenge of estimating baselines in policy gradient methods by comparing responses within a group. Specifically, for each question q, GRPO samples G responses $\{y^{(1)}, \ldots, y^{(G)}\}$ from the current policy and normalizes the rewards to obtain a sequence-level advantage estimate:

$$A^{(i)} = \frac{R_{final}(q, y^{(i)}) - \mu_G}{\sigma_G}, \tag{2}$$

where μ_G and σ_G denote the mean and standard deviation of rewards within the group, respectively. The policy is then updated by maximizing a clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} \min\Big(\rho_t^{(i)} A_t^{(i)},\ \text{clip}\Big(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon \Big) A_t^{(i)} \Big) \right] \tag{3}$$

where $\rho_t^{(i)} = \pi_\theta(y_t^{(i)} \mid q) / \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid q)$ is the per-token importance sampling ratio, and ϵ is the clipping threshold that constrains the policy update to a trust region. All tokens in the trajectory share the sequence-level advantage $A_t^{(i)} = A^{(i)}$. To improve training efficiency, we adopt the duplication strategy proposed in WebSailor [20], where groups with non-zero variance within the batch are duplicated to replace zero-variance groups, as the latter provide no useful training signal.

On-Policy Self Distillation (OPSD). Although GRPO has become the default algorithm for training agents, it has a key limitation in settings with long reasoning chains: relying solely on outcomes lacks process-level supervision. This is especially problematic for incorrect responses, where a small error in the final few tokens may lead to an incorrect answer, yet all preceding tokens receive the same level of penalty. Recent works, including OPSD [54, 63, 13, 38, 57, 35, 17], attempt to alleviate the lack of process-level supervision in long CoT [44] without relying on other teacher models [27]. These methods provide the model with privileged information $c_{privileged}$ as a form of self-teaching, such as reference solutions, environment feedback, or better rollouts, thereby enabling token-level supervision for the student. The training objective is formulated as follows:

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{q\sim\mathcal{D}, y\sim P_S(\cdot|q)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \mathcal{D}(P_T \mid\mid P_S) \right] \tag{4}$$

where $\mathcal{D}$ denotes a divergence measure (e.g., KL divergence), and θ represents the parameters of the student model. The teacher and student distributions can be expressed as:

$$\text{Teacher}: P_T(y \mid q, y_{<t}) = \pi_{\hat{\theta}}(\cdot \mid q, y_{<t}, c_{privileged}) \quad \text{Student}: P_S(y \mid q, y_{<t}) = \pi_\theta(\cdot \mid q, y_{<t}) \tag{5}$$

For training stability, the teacher model's parameters $\hat{\theta}$ are typically not updated through gradient training; common choices include keeping them fixed, updating them via EMA, or synchronizing them with the student model at regular step intervals.

Although significant progress has been made, existing work is largely limited to relatively simple single-turn reasoning tasks. Our work explores how to leverage self-distillation signals to provide more fine-grained supervision for training long-horizon search agents.

3 Step-Level Self-Distilled Policy Optimization

Deep search agents operate through information-seeking actions whose utility emerges only at the level of complete search steps. A retrieval action is valuable not because of individual token choices, but because it effectively localizes uncertainty, gathers relevant evidence, and shapes subsequent exploration. Motivated by this observation, both our privileged information and supervision are aligned with the structure of search steps.

3.1 Privileged Information: Evidence Anchors

In single-turn settings, constructing privileged information is relatively straightforward. For mathematical tasks, many open-source datasets provide reference solutions [63, 10]; for tool use and coding, error messages can be directly obtained [13]; even in the absence of explicit feedback, the highest-scoring rollout among multiple candidates can be selected as privileged teacher information [13]. However, for BrowseComp-style tasks [43], models often require context windows spanning tens or even hundreds of thousands of tokens. In such settings, providing the best rollout as privileged information is impractical, while supplying only the final answer is insufficient to effectively guide multi-step reasoning and action. What is needed is privileged information structured around individual search actions — capable of signaling whether each step retrieves the right evidence.

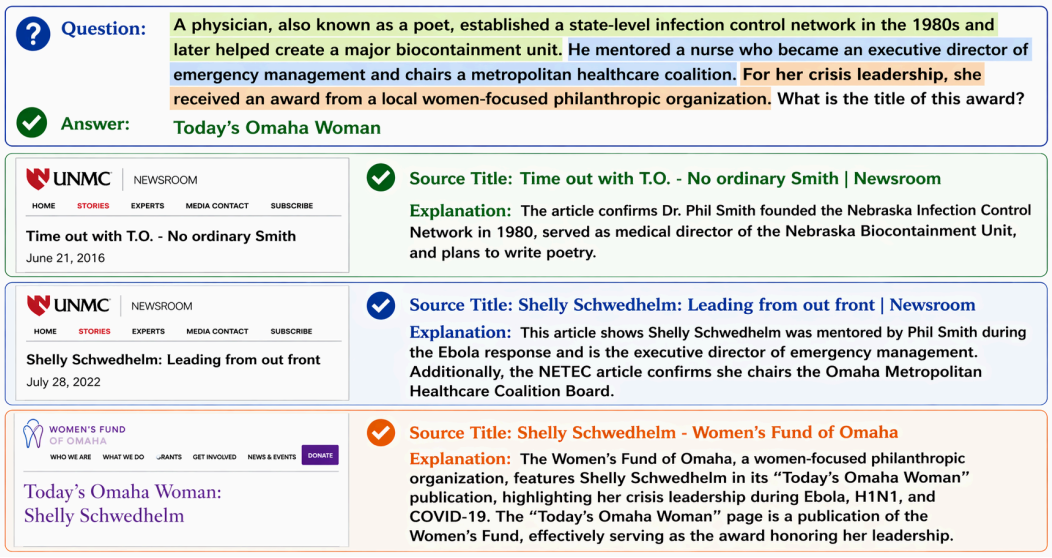


Figure 2: An example question with three evidence anchors. The differently colored evidence anchors correspond to the highlighted conditions in the question.

To construct such privileged information, we construct key pieces of evidence for each QA pair as **Evidence Anchors**. Specifically, we prompt a SOTA LLM to identify as many pieces of evidence as possible that support the ground-truth answer for each question. The prompt we use can be seen in Figure 12. We collect evidence anchors for over 6,000 QA pairs, with an average of 5.24 anchors per question. Figure 2 presents an example from our training data. Detailed statistics and correctness

validation are provided in Appendix G. These privileged signals are incorporated into the teacher model's prompt only for incorrect trajectories, as illustrated in Figure 3. In practice, in addition to evidence anchors, we explicitly provide the teacher with the incorrect answers generated from the student's rollouts, preventing the teacher from repeating the same mistakes in its output distribution.

```
Question: {question}
You previously attempted to obtain the answer {rollout answer}, but it is
    incorrect.
The following are some evidence anchors that may help you:
- {source title}: {explanation}
...
[IMPORTANT]: Do not answer directly from the anchors.
Correctly solve the original question:
```

Figure 3: Teacher prompt template incorporating evidence anchors and previously generated incorrect answers from the student model.

3.2 Step-Level Self-Distilled Advantage Weights

As discussed in prior work [54], directly optimizing Equation 4 can lead to privileged information leakage, which may degrade model performance in the later stages of training. This issue becomes more pronounced in multi-turn settings, where shortcuts introduced by privileged information affect not only the reasoning process itself but also significantly reduce the number of tool calls required for information gathering. To quantify this effect, we sample 50 examples from the training set and compare the cold-start model's accuracy and trajectory length under different information conditions. As shown in Table 1, providing evidence anchors in addition to the question improves accuracy from 46 to 82, while reducing the average number of steps from 17.7 to 3.5. When the final answer is further revealed, accuracy increases to 96, and the average number of steps decreases to 2.4. This substantial discrepancy suggests that the teacher's distribution with privileged information is not an appropriate optimization target for the student. Instead, using self-distillation signals as weighting factors, rather than as gradient directions, helps mitigate this issue [54]. Under this formulation, the update direction remains determined by environment rewards, while the teacher's privileged distribution only modulates the update magnitude, preventing the student from directly fitting an information-asymmetric target. Beyond privileged-information leakage, SRPO [17] further shows that applying OPSD signals only to incorrect trajectories, while preserving the original GRPO objective for correct ones, yields better performance. Applying distillation to already-correct trajectories introduces optimization ambiguity: the model is pushed toward a specific teacher distribution despite having already solved the task, unnecessarily suppressing the diversity of correct solutions. Together, these two insights motivate our method design: we use self-distillation signals as step-level advantage weights rather than optimization targets, and restrict their application to incorrect search-agent trajectories.

	Pass@1	**#Turn**
Question	46	17.7
+ Anchors	82	3.5
+ Answer	96	2.4

Table 1: Comparison of accuracy and average steps before and after providing privileged information.

For each student-generated step τ in trajectory $y^{(i)}$, its thinking tokens $\mathtt{t}_\tau$ and tool-calling tokens $\mathtt{a}_\tau$ jointly constitute one action in the agentic MDP [60]. Under autoregressive factorization, the teacher and student assign the following conditional joint probabilities to this complete step:

$$P_T(\mathtt{t}_\tau, \mathtt{a}_\tau) = \prod_{k \in \tau} \pi_{\hat{\theta}}(y_k \mid c_{\text{privileged}}, q, y_{<k}), \qquad P_S(\mathtt{t}_\tau, \mathtt{a}_\tau) = \prod_{k \in \tau} \pi_\theta(y_k \mid q, y_{<k}). \tag{6}$$

Equation 6 follows standard autoregressive factorization, under which the joint probability of a step is the product of its token-level conditional probabilities. We compute both likelihoods by teacher-forcing the same student-generated step under the corresponding teacher and student configurations: P_T uses the privileged context and teacher parameters, whereas P_S uses the original context and current policy parameters.

We then define the privileged-information gain as the log joint-likelihood ratio of this step action:

$$\Delta_\tau^{step} = \texttt{sg}\left(\log \frac{P_T(\texttt{t}_\tau, \texttt{a}_\tau)}{P_S(\texttt{t}_\tau, \texttt{a}_\tau)}\right), \tag{7}$$

where sg denotes stop-gradient. For tokens y_t belonging to step τ ($t \in \tau$), they share the same step-level advantage weight:

$$w_t \overset{t\in\tau}{=} w_\tau^{step} = \min(\exp(\text{sign}(A^{(i)}) \cdot \Delta_\tau^{step}), 1+\epsilon) \tag{8}$$

where ϵ is a positive hyperparameter, and the $\min$ operation prevents extreme gradient magnitudes. It is important to clarify that trajectories with $R_{final} < 1$ in Equation 1 are treated as incorrect trajectories, since they fail to produce a factually equivalent final answer[2]. Finally, we replace the advantage term in Equation 3 with $\hat{A}_t^{(i)} = w_t A_t^{(i)}$ `if` $R_{final} < 1$ `else` $A^{(i)}$.

As summarized in Table 2, the weight w modulates each token's contribution along two axes: the sign of the trajectory-level advantage $A^{(i)}$ and the step-level teacher–student agreement $\Delta_\tau^{\text{step}}$. For incorrect trajectories with $A^{(i)} < 0$, which represent the typical case for failed rollouts, we use the weight $\min(P_S/P_T,\, 1+\epsilon)$. When the student is overconfident on a step rejected by the teacher ($P_S \to 1$, $P_T \to 0$, i.e., $\Delta_\tau^{\text{step}} < 0$), $w \in (1,\, 1+\epsilon)$ amplifies the penalty, discouraging repeated mistakes. Conversely, when the student is uncertain on a step endorsed by the teacher ($P_S \to 0$, $P_T \to 1$, i.e., $\Delta_\tau^{\text{step}} > 0$), $w \in (0,\, 1)$ reduces the penalty, preserving valid intermediate reasoning within failed rollouts. For incorrect trajectories with $A^{(i)} > 0$, which may occur on challenging questions when a trajectory receives format rewards despite producing an incorrect final answer, we adopt the symmetric weight $\min(P_T/P_S,\, 1+\epsilon)$: a confident teacher paired with an uncertain student amplifies the reward, while a confident student paired with an uncertain teacher dampens it. Across all cases, SSPO adjusts the *magnitude* of updates at step-level resolution without altering the trajectory-level update *direction*.

	$\Delta_\tau^{\text{step}} > 0$	$\Delta_\tau^{\text{step}} < 0$
$A^{(i)} > 0$	$w > 1$ Reward amplified	$w < 1$ Reward dampened
$A^{(i)} < 0$	$w < 1$ Penalty reduced	$w > 1$ Penalty amplified

Table 2: Step-level advantage weight modulation.

4 Experiment

4.1 Experimental Setup

Training Data. Before On-Policy learning, we first followed the WebExplorer [26] pipeline to collect approximately 4,000 English reasoning trajectories with correct final answers and without obvious tool-calling errors, which we used for cold-start initialization. Further details of these trajectories are provided in Appendix D. For On-Policy learning, we sampled around 6,000 English QA pairs from the open-source DeepForge dataset [64], using a difficulty ratio of 1.5:3.5:3.5:1.5 across the four levels. We further employed DeepSeek-V3.2 [9] to construct evidence anchors for each QA pair.

Benchmarks. BrowseComp [43] is a highly challenging information-retrieval benchmark introduced by OpenAI. GAIA [29] is a widely used benchmark for general AI assistants; following WebThinker [22], we evaluate on its text-only subset. FRAMES [16], introduced by Google, is used to assess factual accuracy and reasoning capability. Due to API cost constraints, we evaluate intermediate checkpoints on subsets of BrowseComp and FRAMES, denoted as {BC, FRAMES}-Sub. We adopt the Avg@4 metric to reduce variance, with the temperature set to 0.6 and top-p set to 0.95.

More experimental details could be found in Appendix B.

4.2 Experimental Results

Main Results. Table 3 presents the performance of representative small-scale search agents, alongside our models trained with cold-start initialization, GRPO, and SSPO, as well as the average number of turns required to solve each problem. Despite differences in search scaffolding and the fact that

[2]An incorrect trajectory may still have a positive group-relative advantage, e.g., when it receives the format reward and outperforms other rollouts in the same group. Hence, Equation 8 explicitly considers the advantage sign.

existing agents of comparable size are typically trained on substantially larger datasets, our models achieve competitive—often superior—performance, validating the effectiveness of our training pipeline. Our primary focus, however, is on the performance gains that SSPO brings over GRPO within the same scaffold. Models trained with SSPO consistently outperform their GRPO counterparts by a significant margin. Using the average score across three benchmarks as a representative metric, GRPO achieves a +2.4 improvement over the cold-start baseline, whereas SSPO delivers a markedly larger gain of +4.8. Importantly, these improvements do not come at the expense of reasoning efficiency. The SSPO-trained model requires only a marginally higher number of turns compared to GRPO (20.5 vs. 20.0), indicating that its performance gains are achieved with minimal additional computational cost. As shown in Figure 1 and Figure 10, SSPO demonstrates substantially higher learning efficiency, consistently outperforming GRPO across all three benchmarks at the same number of training steps. To further examine the impact of process supervision, we double the number of training steps for GRPO; notably, SSPO trained for 100 steps already surpasses GRPO trained for 200 steps, highlighting the superior sample efficiency enabled by process-level supervision. Beyond benchmark scores, these gains come with negligible training cost — only about 5% additional overhead from teacher scoring; detailed wall-clock and compute statistics are provided in Appendix E.

Model	**BrowseComp** [43]	**GAIA** [29]	**Frames** [16]	**Average**
Small Size Search Agent (<10B)				
WebSailor-7B [20]	6.7	37.9	–	–
MiroThinker-8B-DPO-v0.1 [42]	8.7	46.6	64.4	39.9
AFM-WebAgent-7B (RL) [21]	5.8	40.8	–	–
DeepDive-9B (RL) [28]	6.3	–	–	–
WebExplorer-8B (RL) [26]	<u>15.7</u>	50.0	<u>75.7</u>	<u>47.1</u>
OffSeeker-8B (DPO) [64]	12.8	<u>51.5</u>	–	–
Our Models (based on Qwen3-8B)				
Our-8B (Cold-Start)	$11.7_{(37.7)}$	$44.8_{(14.9)}$	$67.2_{(7.2)}$	$41.2_{(19.9)}$
Our-8B (GRPO)	$13.6_{(38.0)}$	$47.3_{(14.5)}$	$69.8_{(7.6)}$	$43.6_{(20.0)}$
Our-8B (SSPO)	<u>**15.7**</u>$_{(39.4)}$	**49.3**$_{(14.5)}$	**73.0**$_{(7.7)}$	**46.0**$_{(20.5)}$

Table 3: **Main results on the BrowseComp, GAIA, and Frames benchmarks.** Subscripts indicate the average number of search turns required per problem. <u>Underlined</u> values denote the best overall performance in each column, while **bold** values highlight the best results among our models. Results for other search agents are reported from their respective original publications.

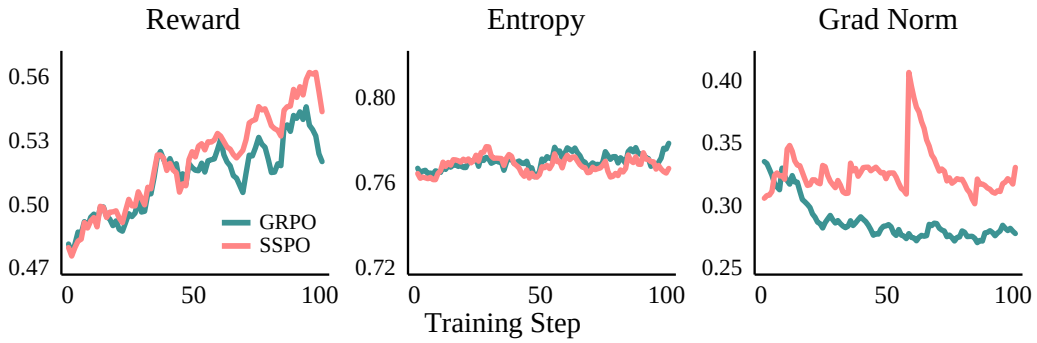


Figure 4: Comparison of training dynamics between GRPO and SSPO (EMA Smoothed).

Training Dynamics. Beyond benchmark scores, we analyze the training dynamics in Figure 4. After an initial warm-up phase of approximately 30–50 steps, SSPO achieves higher rewards than GRPO and maintains this advantage throughout training, indicating more effective optimization in our experiments. Notably, the entropy curves of both methods remain closely aligned, suggesting that

the performance gains of SSPO are unlikely to stem primarily from increased exploration, but rather from more effective policy updates. This observation is further supported by the gradient norm curves: GRPO exhibits a steadily decreasing gradient norm, indicating early saturation of policy updates, whereas SSPO maintains a relatively higher gradient norm throughout training, reflecting more sustained learning signals. Although SSPO shows a transient spike in gradient norm around step 50, this spike quickly subsides and does not appear to harm training stability, as evidenced by the continued improvement in reward. Overall, these results suggest that SSPO improves upon GRPO mainly by enabling more sustained and effective parameter updates, rather than by substantially altering the exploration–exploitation balance.

4.3 Ablation Study

We organize the ablation around three questions: how direct distillation compares with advantage weighting, whether the weighting signal should be defined at token or step level, and what roles the two privileged-information inputs play. Unless otherwise stated, all results are measured on BC-Sub after 50 training steps.

Method	Use of self-distillation signal	Granularity	Evidence Anchors	BC-Sub	Avg. turns
GRPO	None; outcome reward only	–	No	12.8	36.1
OPSD adaptation	Direct optimization target	Token	Yes	11.8	30.9
RLSD/SRPO adaptation	Advantage weighting	Token	Yes	11.8	35.5
Ours (SSPO)	**Advantage weighting**	**Step**	Yes	**14.5**	37.5

Table 4: Comparison of the outcome-only baseline and self-distillation variants after 50 training steps. The RLSD/SRPO adaptation uses the self-distillation signal as token-level advantage weights, following RLSD, and applies it only to incorrect trajectories, following SRPO.

Direct Distillation vs. Advantage Weighting. GRPO provides the outcome-only baseline, reaching 12.8 accuracy with an average of 36.1 turns. Under the same training budget, the direct OPSD adaptation (Appendix F.1) obtains lower accuracy than GRPO (11.8 vs. 12.8) and produces substantially shorter trajectories (30.9 vs. 36.1 turns). Together with Table 1, this pattern is consistent with privileged-information shortcutting: the teacher can solve the task with much less search, and directly matching its distribution transfers this short-search behavior to the student.

The RLSD/SRPO adaptation instead uses the token-level self-distillation signal as advantage weights on incorrect trajectories. Its average trajectory length recovers to 35.5 turns, close to GRPO's 36.1 and substantially above the direct OPSD adaptation's 30.9, while its accuracy remains 11.8. This recovery supports using the teacher signal to modulate the update magnitude rather than as a direct optimization target; however, the unchanged accuracy also shows that token-level weighting alone is insufficient in this setting.

Method	Evidence Anchors	Incorrect feedback	BC-Sub	Sign agreement with full setting (%)	Final-step penalty amplified (%)
GRPO	No	No	12.8	–	–
SSPO w/o Evidence Anchors	No	Yes	12.4	32.8	92.4
SSPO w/o Incorrect Feedback	Yes	No	14.0	81.1	46.7
Ours (SSPO)	Yes	Yes	**14.5**	100.0	86.3

Table 5: Ablation of the two privileged-information sources. BC-Sub results are measured after 50 training steps. The two rightmost columns are computed by rescoring 512 incorrect trajectories, using the sign under the full setting as a reference. "Final-step penalty amplified" denotes the proportion of trajectories for which the final-step weight is greater than one.

Step-Level Rather Than Token-Level. We then isolate supervision granularity by holding advantage weighting, Evidence Anchors, and incorrect-trajectory routing fixed. The token-level RLSD/SRPO adaptation assigns a separate weight to every token (Appendix F.2), whereas SSPO shares one joint-probability weight across the reasoning and tool-calling tokens that form a complete information-seeking step. This controlled change improves BC-Sub accuracy from 11.8 to 14.5 while maintaining

the average trajectory length (35.5 vs. 37.5 turns). The entropy curves in Figure 5 provide complementary evidence: token-level weighting produces steadily increasing entropy, whereas step-level weighting remains more stable. These results support aligning the supervision granularity with the complete search action rather than fragmenting it across individual tokens.

Roles of Evidence Anchors and Incorrect-Answer Feedback. Table 5 separates the two sources of privileged information. Incorrect-answer feedback alone does not improve over GRPO (12.4 vs. 12.8), whereas Evidence Anchors alone improve accuracy to 14.0. Combining both yields the best result, 14.5, showing that Evidence Anchors provide the primary process-supervision signal and incorrect-answer feedback is complementary.

To examine how they contribute to the full step score, we rescore the same 512 incorrect trajectories under each single-information setting. This is a decomposition of the deployed signal rather than an assumption that the full setting is an unbiased ground-truth label. Evidence Anchors alone agree with the sign of the full signal on 81.1% of steps, compared with 32.8% for incorrect-answer feedback alone, indicating that Evidence Anchors primarily determine the evaluation of intermediate information-seeking actions. Conversely, incorrect-answer feedback amplifies the final-step penalty in 92.4% of trajectories, compared with 46.7% under Evidence Anchors alone. Thus, the incorrect answer primarily identifies the failed terminal conclusion, while Evidence Anchors ground credit assignment across the preceding search process.

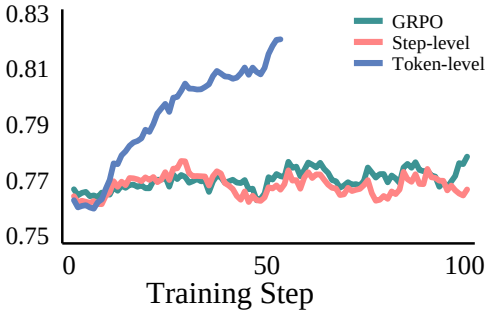


Figure 5: Entropy curves of GRPO, step-level, and token-level supervision during training.

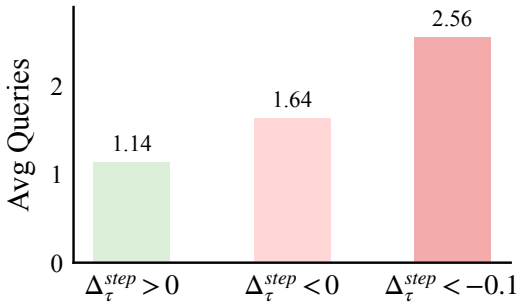


Figure 6: Average number of queries under different step-level privileged-information gains.

5 Case Study: Which Steps are Exempted from Penalty?

To better understand why SSPO is effective, we analyze how the teacher assigns penalties across different types of steps. We find that **the teacher does not simply reward steps based on their lexical overlap with evidence anchors. Instead, it modulates penalties according to whether a step performs a focused and informative verification action**.

Specifically, steps that receive reduced penalties are typically compact, well-scoped queries targeting a single local constraint implied by the evidence anchors, as shown in Figure 7. Although these steps may contain fewer anchor words, they effectively reduce uncertainty along the reasoning chain. In contrast, steps that receive stronger penalties are often long, diffuse queries that concatenate multiple surface terms from the question or anchors without forming a precise information need, as shown in Figure 8. Despite their high lexical overlap with anchors, such steps receive amplified penalties, indicating that the teacher prioritizes information quality over word matching. This behavior is further supported by quantitative analysis. Figure 6 shows the relationship between the average number of queries in search tool calls and the privileged-information gain. Steps with positive privileged-information gain $\Delta_\tau^{step} > 0$ use significantly fewer queries than those with negative gain $\Delta_\tau^{step} < 0$. Overall, SSPO encourages targeted, evidence-grounded information-seeking by reducing penalties for precise verification steps while amplifying penalties for broad and unfocused exploration.

6 Conclusion

We presented SSPO, a step-level self-distillation method that extends On-Policy Self-Distillation from single-turn reasoning to multi-turn deep search agents. To construct a meaningful self-teacher

```
<think>
Perhaps the specific ERNI employee referenced is not the person listed in the
     program committee (Florian, Dominik etc). The above LinkedIn snippet for
     Patrick Pfeifer shows "iSAQB - Foundation Level (CPSA-F)", which matches
     the certification.
Now does Patrick Pfeifer specialize in medical technology software?
    Potentially, as ERNI works in medtech, and "Patrick Pfeifer" might be
    focusing on medical solutions, but we need verification.
Search "Patrick Pfeifer ERNI medical technology".
</think>
<tool_call>
"search queries": ["Patrick Pfeifer ERNI medical technology"]
</tool_call>
```

Figure 7: $\Delta_\tau^{step} > 0$: compact single-query verification targeting one specific entity-attribute relation.

```
<think>
ERNI matches the consultancy with offices across Europe and Asia.
Now find an ERNI employee who is a "medical technology software specialist"
    and holds a "foundational certification in software architecture". The
    foundation certification could refer to the iSAQB CPSA-F certification.
We also have that this employee serves on the conference committee for
    BaselOne and recently co-founded an initiative linking local industry and
     academia.
The program committee includes Ix-chel Ruiz, Dominik Berger, Marcus Fihlon,
    Robin Mezger, Florian Stampe, Frank Baier, etc. Among these, perhaps one
    is a medical technology software specialist.
But we could also search for "BaselOne conference committee ERNI".
</think>
<tool_call>
"search queries": [
  "BaselOne conference committee ERNI",
  "ERNI medical technology software architect BaselOne",
  "ERNI medical technology software specialist BaselOne committee",
  "CPSA Foundation ERNI employee"]
</tool_call>
```

Figure 8: $\Delta_\tau^{step} < 0$: diffuse multi-query search with high lexical overlap but poor localization.

in a setting where neither reference solutions nor execution feedback are available, we introduced Evidence Anchors—structured, step-aligned privileged information whose granularity matches that of the search process itself. From this teacher, SSPO derives a self-distillation signal, recasts it as step-level advantage weights, and applies it only to incorrect trajectories, thereby injecting fine-grained process supervision while sidestepping both privileged-information leakage and the ambiguous gradients that direct distillation would otherwise impose on already-correct rollouts.

Across BrowseComp, GAIA, and FRAMES, SSPO surpasses GRPO trained for twice as many gradient steps, while sustaining higher rewards and more stable gradient norms throughout optimization. Beyond the aggregate gains, our analysis reveals a coherent shaping of the policy update: penalties contract on compact, evidence-grounded verification steps and expand on diffuse, underspecified queries. This suggests that step-level self-distillation guides the agent toward more targeted and efficient information-seeking, rather than simply lifting final-answer accuracy.

Acknowledgments and Disclosure of Funding

Funding in direct support of this work: NeurIPS 2026 Financial Aid (travel/registration support).

Competing interests: The authors declare no competing interests.

References

[1] Zhipu AI. Glm-4.6. https://docs.z.ai/guides/llm/glm-4.6, 2025.

[2] Zhipu AI. Glm-5.1. https://docs.z.ai/guides/llm/glm-5.1, 2026.

[3] Anthropic. Introducing Claude 4 — anthropic.com. https://www.anthropic.com/news/claude-4, 2025. [Accessed 19-03-2026].

[4] Anthropic. Introducing claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026.

[5] Ziqian Bi, Keyu Chen, Chiung-Yi Tseng, Danyang Zhang, Tianyang Wang, Hongying Luo, Lu Chen, Junming Huang, Jibin Guan, Junfeng Hao, Xinyuan Song, and Junhao Song. Is gpt-oss good? a comprehensive evaluation of openai's latest open source models, 2025.

[6] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 10818–10838. PMLR, 2025.

[7] Google Deepmind. Gemini 3.1 pro: Best for complex tasks and bringing creative concepts to life. https://deepmind.google/models/gemini/pro/, 2026.

[8] DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

[9] DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, et al. Deepseek-v3.2: Pushing the frontier of open large language models, 2025.

[10] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, et al. Openthoughts: Data recipes for reasoning models, 2025.

[11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025.

[12] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[13] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

[14] Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O Arik, and Jiawei Han. An empirical study on reinforcement learning for reasoning-search interleaved llm agents. *arXiv preprint arXiv:2505.15117*, 2025.

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

[16] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation, 2024.

[17] Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing, 2026.

[18] Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution, 2026.

[19] Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning, 2025.

[20] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025.

[21] Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, He Zhu, Dingfeng Shi, Piaohong Wang, Yeyi Guan, Xiangru Tang, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Jiaheng Liu, Ge Zhang, and Wangchunshu Zhou. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl, 2025.

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

[23] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

[24] Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. 2025.

[25] Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, Yunan Huang, Mozhi Zhang, Pengyu Zhao, Junjie Yan, and Junxian He. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond, 2025.

[26] Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. Webexplorer: Explore and evolve for training long-horizon web agents, 2025.

[27] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

[28] Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl, 2025.

[29] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023.

[30] MiniMax. Minimax m2.7: Early echoes of self-evolution, 2026.

[31] OpenAI. Gpt-5-nano. `https://developers.openai.com/api/docs/models/gpt-5-nano`, 2025.

[32] OpenAI. Introducing gpt-oss. `https://openai.com/index/introducing-gpt-oss/`, 2025.

[33] OpenAI. Introducing gpt-5.5. `https://openai.com/index/introducing-gpt-5-5/`, 2026.

[34] Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026.

[35] Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation, 2026.

[36] Florian Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167. Association for Computational Linguistics, 2019.

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[38] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026.

[39] Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026.

[40] Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025.

[41] Kimi Team. Kimi k2.6: Advancing open-source coding. `https://www.kimi.com/blog/kimi-k2-6`, 2026.

[42] MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Xuan Dong, et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025.

[43] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025.

[44] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[45] Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents, 2026.

[46] Haoze Wu, Cheng Wang, Wenshuo Zhao, and Junxian He. Mirage or method? how model-task alignment induces divergent rl conclusions, 2025.

[47] Haoze Wu, Yunzhi Yao, Wenhao Yu, and Ningyu Zhang. Recode: Updating code api knowledge with reinforcement learning, 2025.

[48] Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification, 2026.

[49] X.AI. Grok 4.1 fast and agent tools api. `https://x.ai/news/grok-4-1-fast`, 2025.

[50] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, 2025.

[51] Peiran Xu, Zhuohao Li, Xiaoying Xing, Guannan Zhang, Debiao Li, and Kunyu Shi. Principle process reward for search agents, 2026.

[52] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation, 2026.

[53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, et al. Qwen3 technical report, 2025.

[54] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr, 2026.

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

[56] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[57] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026.

[58] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025.

[59] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025.

[60] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026. Survey Certification.

[61] Yabo Zhang, Yihan Zeng, Qingyun Li, Zhen Hu, Kavin Han, and Wangmeng Zuo. Tool-r1: Sample-efficient reinforcement learning for agentic tool use, 2025.

[62] Yaocheng Zhang, Haohuan Huang, Zijun Song, Yuanheng Zhu, Qichao Zhang, Zijie Zhao, and Dongbin Zhao. Criticsearch: Fine-grained credit assignment for search agents via a retrospective critic, 2025.

[63] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026.

[64] Yuhang Zhou, Kai Zheng, Qiguang Chen, Mengkang Hu, Qingfeng Sun, Can Xu, and Jingjing Chen. Offseeker: Online reinforcement learning is not all you need for deep research agents, 2026.

A Related Work

A.1 Reinforcement Learning with Verifiable Rewards (RLVR)

Building on works such as DeepSeek-MATH [37] and DeepSeek-R1 [11], RLVR has become a standard component in LLM training pipelines and is widely regarded as essential for improving reasoning capabilities across domains, including but not limited to mathematics [59, 58], logical reasoning [50, 25, 46], coding [24, 47], and verifiable agent tasks [14, 15, 61]. Although RLVR is considerably more complex to implement than SFT, its On-Policy nature confers a qualitatively distinct advantage: by exploring the environment during training, the model is not confined to imitating fixed teacher trajectories and can autonomously discover effective reasoning strategies. DeepSeek-R1 [11] demonstrates that, without any process-level supervision, RL training can spontaneously elicit sophisticated behaviors such as backtracking and self-reflection—phenomena collectively referred to as "aha moments." This stands in contrast to SFT, which by directly fitting a static teacher distribution fundamentally forecloses such exploration [36, 6, 48, 38].

A.2 Process Rewards in Search Agent Training

To address the sparsity of outcome-only rewards in long-horizon agent settings, several recent works introduce process rewards that evaluate intermediate steps during search. CriticSearch [62] employs a frozen asymmetric critic LLM that retrospectively evaluates each interaction turn using privileged information from the complete trajectory and gold answers, converting these assessments into dense turn-level rewards for policy optimization. PPR [51] trains a dedicated principle-based process reward model that grounds step-wise judgments in interpretable principles such as correctness, relevance, and consistency, and further introduces a reward normalization strategy to balance local process fidelity against global task success. SmartSearch [45] takes a query-centric view, designing a dual-level credit assessment mechanism that scores each intermediate search query for both novelty and usefulness, and uses these scores to selectively refine low-quality queries via a separately trained smaller model.

While these approaches demonstrate the value of fine-grained supervision for search agents, they share two notable limitations. First, their experimental benchmarks are primarily composed of standard multi-hop QA tasks, such as HotpotQA [55] and 2WikiMultiHop [12], which involve relatively shallow retrieval chains. None of them evaluate on BrowseComp-style tasks that require navigating dozens of search and browse steps to resolve highly constrained, multi-conditional queries—precisely the setting where per-step credit assignment is most critical. Second, all three methods rely on a separate LLM as a per-step evaluator, which introduces additional inference cost and a dependency on the quality and calibration of that external scorer. In contrast, SSPO eliminates the need for an explicit step-level reward model by converting self-distillation signals directly into step-level advantage weights, making fine-grained process supervision both practically lightweight and tightly integrated with the on-policy training objective.

A.3 On-Policy (Self-)Distillation

On-Policy Distillation (OPD) [27, 39, 23, 52] improves upon traditional off-policy supervised fine-tuning (SFT) by allowing the student to generate its own rollouts while using teacher logits for supervision, thereby reducing the distribution mismatch between training and inference. However, OPD still relies on a teacher model that operates in a compatible vocabulary space and is strictly more capable than the student, which limits its practical applicability.

A growing body of work seeks to remove this dependency by constructing self-teachers from privileged information or environmental feedback. Self-Distilled Reasoner [63] shows that On-Policy Self-Distillation (OPSD), when augmented with reference solutions as privileged prefixes, achieves strong performance on single-turn mathematical reasoning tasks. RL via Self-Distillation [13] extends this paradigm to tool-use and coding scenarios, where environment feedback—such as compiler errors or execution outputs—serves as a form of privileged supervision. Self-Distilled RLVR (RLSD) [54] further identifies a critical limitation of direct OPSD: optimizing the student toward a teacher conditioned on privileged information can introduce information leakage, where the policy implicitly exploits signals unavailable at test time. To address this issue, RLSD [54] proposes converting distillation signals into advantage weights rather than directly using them as gradient targets, thereby preserving the on-policy reward as the primary optimization objective. SRPO [17] further observes that applying self-distillation uniformly across both correct and incorrect trajectories introduces

ambiguous optimization signals, and instead advocates restricting OPSD signals to incorrect trajectories. CRISP [35] explores iterative self-policy distillation for compressing reasoning chains, while On-Policy Context Distillation [57] studies distillation from context-augmented teachers in language model settings. More broadly, self-distillation has also been shown to support continual learning without catastrophic forgetting [38].

Our work builds on these insights and extends OPSD to multi-turn deep search agents, a setting that introduces two key challenges not addressed by prior work: (1) how to construct meaningful privileged information for open-ended information retrieval tasks, where neither reference solutions nor execution feedback are directly available; and (2) how to define an appropriate supervision granularity that aligns with the natural unit of search behavior — information-seeking actions — rather than defaulting to token-level signals.

B More Experimental Details

Scaffold Details. Our scaffold is based on the ReAct paradigm [56], enabling interaction with two tools: `search` and `browse` (see Appendix C). We use no specialized system prompts during either training or evaluation, providing only the necessary tool descriptions. Except for the teacher model, the user prompt contains only the question itself. The model is required to invoke a tool at each step, except for the final step where it outputs the answer, until termination or reaching the step limit. If no valid tool call is produced at any step, the trajectory is terminated and scored according to the final response. Following prior work [26, 20], we use an LLM judge to evaluate answer correctness.

Training Details. We use Qwen3-8B [53] as the base model for all experiments. For cold-start Supervised Fine-Tuning (SFT), we set the batch size to 32 and the learning rate to 1×10^{-5}, with linear warmup followed by cosine decay, and train the model for 1k steps. For On-Policy learning, we train on approximately 6k samples using a fixed learning rate of 1×10^{-6} and a batch size of 64, with 8 rollouts per question. For our method, we set $\epsilon = 0.2$, and reinitialize the teacher model with the current policy model every 50 training steps. Throughout training, we set the maximum context length to 128K and the maximum number of agent steps to 100.

Loss Masking. For both SFT and On-Policy learning, we compute the training loss only on agent-generated tokens, including the Thought and Action segments, as well as the final answer when applicable. Tokens returned by the environment, such as search results and browsed page contents, are provided only as context for subsequent agent decisions and are masked out from the loss. This is because observation tokens are generated by external tools rather than by the agent policy, and therefore should not be predicted or optimized as part of the agent's action distribution.

C Agent Tool Schemas

Similar to WebExplorer [26], we provide the LLM with two tools for information retrieval:

Search Tool. Our search tool enables the LLM to issue multi-keyword queries simultaneously. It leverages the Serper API[3] to return relevant information triplets `(title, URL, snippet)` to the LLM.

[3] https://serper.dev/

Search Tool Schema

```
type: function
function:
  name: search
  description: Web search.
  parameters:
    type: object
    properties:
      queries:
        type: array
        description:
          The queries will be sent to Google via Serper API. You will
          get the brief search results with (title, url, snippet)s for
          each query.
        items:
          type: string
    required: queries
```

Browse Tool. Our browse tool retrieves and processes content from specific URLs using content extraction and language model capabilities. Specifically, content extraction is powered by Jina's service[4], while long-context retrieval is handled by another LLM with a context window exceeding 256k tokens.

Browse Tool Schema

```
type: function
function:
  name: browse
  description: Explore specific information in a url.
  parameters:
    type: object
    properties:
      url:
        type: string
        description:
          The url will be browsed, and the content will be sent to a
          Large Language Model (LLM) as the based information to
          answer a query.
      query:
        type: string
        description:
          The query to this url content.
    required: [url, query]
```

D Cold-Start Trajectory Collection

D.1 QA Generation

To obtain high-quality teacher trajectories for cold-start initialization, we first require sufficiently challenging QA pairs [20, 19, 40, 26, 22]. However, for a long time, such data has been largely lacking in the open-source community. WebExplorer [26] proposes a model-based approach for QA pair synthesis, enabling the generation of sufficiently challenging examples without constructing large-scale web topology graphs. It divides QA pair synthesis into two stages: *Model-Based Exploration* and *Iterative Query Evolution*, and employs the costly Claude model in both stages. When using other, lower-cost API models, the difficulty of the generated questions tends to drop significantly, as shown in Table 6.

[4] https://jina.ai/

Model	Accuracy	#Tool Calling	#Search	#Browse
Claude-4-Sonnet [3]	**58.3**	**20.2**	**15.0**	**5.2**
Grok-4.1-Fast [49]	81.7	12.5	8.0	4.5
GLM-4.6 [1]	75.5	12.5	8.2	4.3
DeepSeek-V3.2 [9]	80.0	8.9	6.2	2.7
DS-V3.2 + Modified Prompt	62.0	15.0	11.1	3.9

Table 6: Comparison of question difficulty across different models, evaluated by accuracy and the average number of tool calls (measured on GPT-5-Nano [31]).

We find that the prompt used in the second stage is somewhat too simplistic, making it difficult for models with weaker instruction-following ability to effectively increase the difficulty of the generated questions. We revise it to the version shown in Figure 9, which substantially improves question difficulty even when using DeepSeek-V3.2 [9] under the same setting. As shown in Table 6, on 100 generated instances, the accuracy of GPT-5-Nano [31] decreases from 80.0 to 62.0, while the average number of tool calls increases from 8.9 to 15. We then use the improved pipeline with the lower-cost DeepSeek-V3.2 [9] to generate over 6,000 QA pairs.

D.2 Teacher Trajectories

With high-quality QA pairs in hand, we then employ a teacher model to generate ReAct-style reasoning trajectories [56]. We deploy GPT-OSS-120B [32] as the teacher model and set the reasoning effort to high. This setup yields the model's native long-form reasoning trajectories, rather than user-facing compressed CoTs [44]. In addition, when comparing trajectories from GPT-OSS-120B [32] with those from other commercial flagship models, we observe two key characteristics: (1) Compared to flagship models, GPT-OSS-120B [32] makes more tool calls. (2) When both browse and search tools are available, GPT-OSS-120B [32] invokes browse to retrieve webpage content significantly more frequently than other models.

Model	#Tool Calling	#Search	#Browse	#Browse Frac
Claude-4-Sonnet [3]	5.67	4.48	1.19	21.0%
GLM-4.6 [1]	4.92	3.87	1.05	21.3%
GPT-OSS-120B [32]	**18.14**	11.90	6.24	**34.4%**

Table 7: Comparison of Tool Usage Across Different Models.

As shown in Table 7, for the same queries, GPT-OSS-120B [32] performs more than three times as many tool calls as other models and more frequently invokes the browse tool to retrieve detailed information from the web. This highlights differences in reasoning strategies across models: flagship models often narrow down candidate answers to a very small set using internal knowledge in the first step, requiring only minimal verification. In contrast, GPT-OSS-120B [32] does not exhibit similarly rich internal knowledge in domains such as the social sciences and arts [5], making its trajectories more suitable for student models that lack sufficient internal knowledge. Although we use only 4,000 trajectories—significantly fewer than the 13,000 used in WebExplorer [26], we achieve performance on par with, or even surpassing, WebExplorer-8B-SFT [26]. This highlights the advantage of GPT-OSS-120B [32] as a teacher model.

Table 8 presents the statistics of the final teacher trajectories. We ultimately retain only correct trajectories and remove those with obvious tool-calling errors. In addition, we observe that incorrect trajectories are approximately three times longer than correct ones, underscoring both the necessity and urgency of fine-grained supervision for erroneous trajectories.

```
You must **significantly increase** the difficulty of the following question
    while ensuring the correct answer remains **uniquely identifiable**.

Original question: {orig_question}
Original truth: {truth}

You **must aggressively** apply ALL of the following strategies to make the
    question harder:
1. **Remove highly specific clues**: delete dates, numbers, full names,
    institutions, awards, locations, publication names, etc.; keep only the
    minimal signals needed for unique identification
2. **Blur and generalize**: replace concrete information with uncertain or
    approximate descriptions, while keeping the truth uniquely resolvable
3. **Add distractor-like similar entities**: introduce misleading cues so
    shallow reasoning fails and deeper inference is required
4. **Refer indirectly to the entity**: use less-common identifiers---
    predecessor/successor relationships, indirect influence, obscure nicknames,
     abstract impact, associated figures, etc.
5. **Iterative escalation**: perform **5 evolution steps**; each step must
    remove or obscure at least one previously clear attribute, and be **
    strictly harder** than the previous version
6. Ensure **uniqueness of the final truth**: despite the ambiguity, the
    question must still map to the exact same truth

You may use search and browsing tools to verify uniqueness during rewriting.

---

Output format:

For each iteration:
<question>
{more difficult question version}
</question>

After the 5th iteration, output the final result:
<answer>
<question>{the most difficult version}</question>
<truth>{the exact same truth}</truth>
</answer>

Do NOT include any other explanations, comments, or formats.
If any iteration fails to increase difficulty, the task is considered failed.
```

Figure 9: Modified Prompt template used in the *Iterative Query Evolution* stage.

	Accuracy	**#Tool**	**#Browse**	**#Search**
All	0.74	32.1	10.5	21.6
Correct	-	20.5	8.3	12.2
Incorrect	-	65.5	16.9	48.6

Table 8: Statistics of the teacher trajectories.

E More Experimental Results

Training Dynamics. As shown in Figure 10, SSPO consistently outperforms GRPO across all three benchmarks throughout training, and achieves stronger performance with fewer training steps, indicating both higher sample efficiency and better final performance.

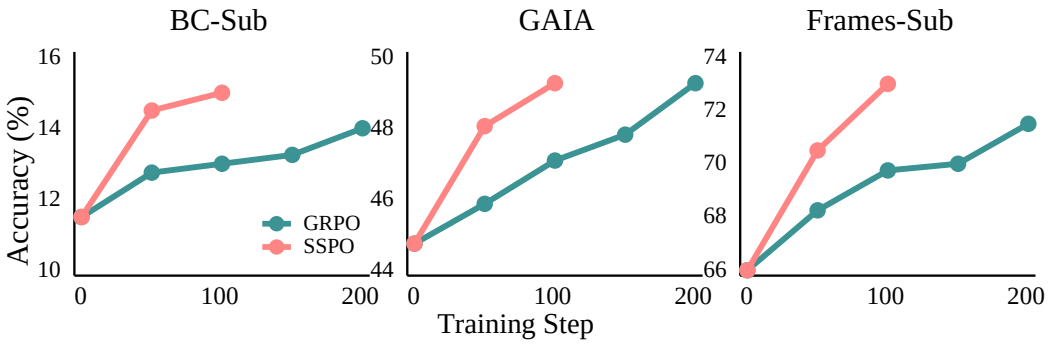


Figure 10: Training dynamics of GRPO and SSPO on BC-Sub, GAIA, and Frames-Sub.

Acceptable Training Overhead. Although SSPO achieves significant performance gains, it requires one additional forward pass to obtain the teacher logits. When training on two 8-accelerator nodes with 140 GB memory each, the time breakdown across different stages of each training step is reported in Table 9. The additional teacher forward pass accounts for only ∼5% of the total step time, making the overhead negligible relative to the performance gains.

	Collect Traj	Update Actor	Policy logP	Teacher logP	Other
Proportion (%)	59.2	23.3	5.5	6.5	5.5

Table 9: The proportion of time consumed by different components within each training step.

F More Details about Ablation Study

F.1 Training Objective of OPSD

We replace the advantage $A_t^{(i)}$ in Equation 3 with $\log P_T(y \mid c_{privileged}, q, y_{<t}) - \log P_S(y \mid q, y_{<t})$ as the training objective of OPSD. This objective is more tractable than computing the KL divergence over the full vocabulary and has been widely shown to be effective [27].

F.2 Token-Level Advantage Weights

Similar to Step-Level Self-Distilled Advantage Weights in Section 3.2, we can compute the privileged-information gain for each token $y_t \in y^{(i)}$ as:

$$\Delta_t^{token} = \texttt{sg}(\log P_T(y_t, |c_{privileged}, q, y_{<t}) - \log P_S(y_t|q, y_{<t})) \tag{9}$$

We can then compute the advantage weight for each token:

$$w_t = \min(\exp(\text{sign}(A^{(i)}) \cdot \Delta_t^{token}), 1 + \epsilon) \tag{10}$$

Finally, we replace the advantage term with $\hat{A}_t^{(i)} = w_t A_t^{(i)}$ `if` $R_{final} < 1$ `else` $A^{(i)}$.

G More Details about Evidence Anchors

Evidence Anchor Quality Validation. As shown in Figure 12, during Evidence Anchor collection, we also require the model to provide the URL of each source page. Although URLs themselves contain limited instructional semantic information and are therefore not included in the teacher prefix, we use them for automatic quality validation. Specifically, to reduce the risk that the LLM fabricates non-existent sources, we use Jina to access each provided webpage and verify both URL accessibility and whether the retrieved page title matches the source title reported by the LLM. Encouragingly, thanks to recent improvements in LLM capability and our use of *search* and *browse* tools during anchor collection, only a very small fraction of Evidence Anchors contain inaccessible URLs. We filter out these invalid anchors and remove QA pairs containing invalid Evidence Anchors from the training set.

Evidence Anchor Statistics. Figure 11 shows the distribution of the number of evidence anchors per question in our training data. The distribution is centered around 5 anchors per question, with a mean of 5.24. Most questions contain between 4 and 6 anchors, which together account for the majority of the dataset. The frequency drops off on both sides, with very few questions having fewer than 3 or more than 8 anchors. This indicates that our data construction process produces moderately sized evidence sets, balancing coverage and conciseness for effective supervision.

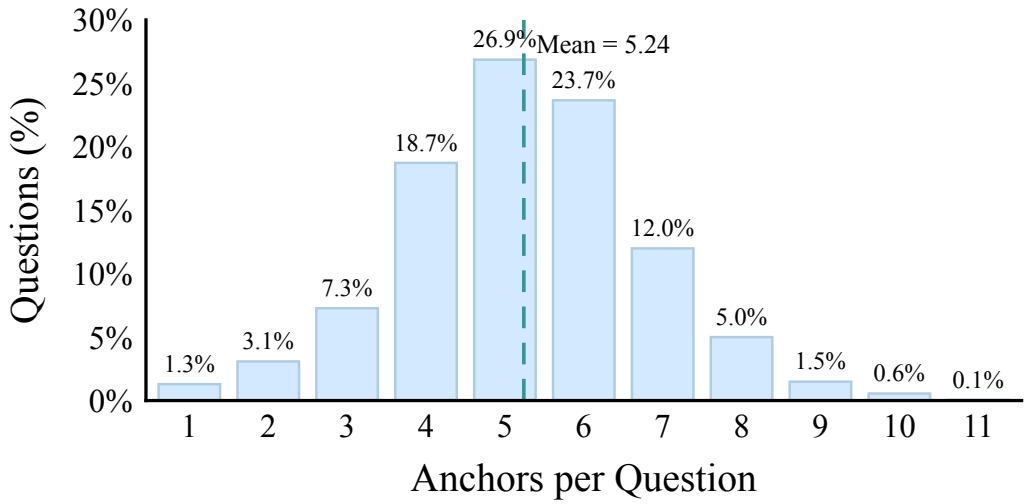


Figure 11: Statistics of Evidence Anchor Counts.

H Limitations and Future Work

Due to the high cost associated with API usage (e.g., Serper, Jina, and LLM services), the scale of both data construction and experimental evaluation is constrained. For both cold-start and on-policy training, we rely on only a few thousand samples. Moreover, due to computational and time limitations, our experiments are restricted to 8B-scale models, and we do not evaluate performance on larger model sizes. Additionally, due to the English-only chain-of-thought characteristics of the teacher model (GPT-OSS-120B [32]), both training and evaluation are confined to English, without incorporating multilingual data such as Chinese or Japanese. This may limit the generality of our findings to broader multilingual settings.

In future work, we aim to investigate the role of fine-grained supervision in improving existing RLVR methods across more diverse agent scenarios. We also plan to scale up both the data and experimental scope, including exploring larger model sizes and extending the framework to multilingual settings.

```
CRITICAL ROLE: You are a strict Verification Engine. Your sole mission is to
    find evidence that the [Standard Answer] SPECIFICALLY satisfies each
    constraint of the [Question].
STRATEGIC SEARCH MANDATE:
1. DECOMPOSE: Break the [Question] into independent, verifiable conditions.
2. ANCHORING SEARCH: For each condition, your search query MUST include the [
    Standard Answer] or be related to the [Standard Answer].
   - BAD Search: "who is the COOP leader in Amarillo" (This is solving)
   - GOOD Search: "Angela Margrave National Weather Service Amarillo COOP
    leader" (This is verifying)
3. PROVE SATISFACTION: A condition is only 'Verified' if you find a source that
     explicitly links the [Standard Answer] to that specific requirement.
OUTPUT FORMAT (Strict XML):
<evidences>
  <evidence>
    <condition>The specific requirement/condition extracted from the [Question
    ]</condition>
    <source_title>The title of the source web page or article</source_title>
    <source_url>The full URL of the supporting web page</source_url>
    <explanation>Briefly explain how this evidence explicitly proves that the [
    Standard Answer] satisfies this specific condition</explanation>
  </evidence>
</evidences>
[Example]:
Question: A U.S. citizen science initiative relies on volunteers to collect
    daily meteorological data. The data gathered is vital for a federal agency
     operating within the same department as the body responsible for mapping
    the ocean floor. The forecast office for a region known for a major city
    with a famous public art installation of partially buried cars has a
    specific leader for this program. What is the name of this program leader?
Answer: Angela Margrave

Your Output:
<evidences>
  <evidence>
    <condition>The forecast office must be located in a region known for a
    major city with a famous public art installation of partially buried cars
    .</condition>
    <source_title>Cadillac Ranch - Wikipedia</source_title>
    <source_url>https://en.wikipedia.org/wiki/Cadillac_Ranch</source_url>
    <explanation>The evidence shows that Cadillac Ranch is a famous art
    installation of half-buried cars located in Amarillo, Texas. This proves
    the required forecast office region is Amarillo.</explanation>
  </evidence>
  <evidence>
    <condition>The specific leader for this meteorological program at the
    identified forecast office (Amarillo) must match the answer.</condition>
    <source_title>National Weather Service - COOP Recruitment Amarillo</
    source_title>
    <source_url>https://www.weather.gov/ama/COOP_Recruitment</source_url>
    <explanation>The official NWS page explicitly lists Angela Margrave as the
    COOP Program Leader at the Amarillo, TX forecast office, confirming the
    standard answer perfectly satisfies the final requirement.</explanation>
  </evidence>
</evidences>

[Question]: {question}
[Answer]: {answer}
```

Figure 12: Prompt template used to collect Evidence Anchors.

I Societal Impacts

This work studies training methods for language-model agents that browse the open web to answer questions. We see both potential benefits and risks.

Potential benefits. More sample-efficient training of deep search agents can reduce the compute needed to reach a given level of reliability on multi-step information-seeking tasks. Better credit assignment may also improve faithfulness of intermediate search and reading steps, which matters for research assistance, education, and other settings where users need not only a final answer but a traceable path of evidence.

Potential risks and limitations. The same capabilities can be misused: automated agents may scrape content at scale in ways that violate site terms of use, amplify low-quality or misleading sources, or assist social-engineering and privacy-invasive information gathering. Our method improves how agents learn from outcome feedback; it does not by itself add safety filters, access controls, or content policies. Experiments are limited to public question-answering benchmarks and do not evaluate dual-use or adversarial settings.

Mitigation. We release code and data under standard research licenses and encourage deployment behind appropriate rate limits, logging, and human oversight. We recommend evaluating search agents for harmful query patterns and source reliability before real-world use. We do not claim that SSPO alone resolves these risks.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions, including Evidence Anchors, step-level self-distillation, and empirical gains on deep search benchmarks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations and Future Work section discussing constraints in data scale due to API costs, limited evaluation on 8B-scale models, and restriction to English-only settings, as well as their implications for generalization and future extensions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide the training setup, datasets, evaluation protocol, model backbone, reward design, hyperparameters, and implementation details needed to reproduce the main experiments. Code and data are publicly available: code at `https://github.com/hkust-nlp/SSPO` and the associated dataset at `https://huggingface.co/datasets/WaitHZ/SSPO-data`.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: Code and data are now publicly available, rather than withheld for review. Code with instructions to reproduce the main experiments is at `https://github.com/hkust-nlp/SSPO`. The dataset is at `https://huggingface.co/datasets/WaitHZ/SSPO-data`. The paper also describes the training setup, data construction, evaluation protocol, hyperparameters, and implementation details.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper reports training data, benchmarks, model backbone, optimization settings, rollout configuration, and evaluation metrics.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: We report evaluation results using averaged metrics (e.g., Avg@k) to reduce variance in long-horizon agent evaluation.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the compute resources used for training and evaluation, including GPU type, context length, rollout settings, and estimated compute cost.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics and uses public benchmarks and web-accessible information without human-subject experiments.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Yes. We discuss both potential positive and negative societal impacts, together with mitigation strategies, in Appendix I.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained model or scraped dataset requiring special safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: We cite the existing datasets, models, and tools used in the paper and report their licenses or terms when available.

Guidelines:

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

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces new assets in the form of evidence anchors constructed for QA pairs. The paper documents their construction process, including data sources, prompt templates, filtering procedures, and generation pipeline. These assets are released with the public code and data (`https://github.com/hkust-nlp/SSPO`, `https://huggingface.co/datasets/WaitHZ/SSPO-data`), which include the documented construction artifacts.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject studies.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject studies, so IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs are a core component of the method, including QA generation, evidence-anchor construction, teacher trajectory generation, and automatic answer judging, and these uses are described in the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.