# WebAnchor: Anchoring Agent Planning to Stabilize Long-Horizon Web Reasoning

**Xinmiao Yu[1,2], Liwen Zhang[1], Xiaocheng Feng[2], Yong Jiang[1], Bing Qin[2], Pengjun Xie[1], Jingren Zhou[1],**

[1]Tongyi Lab, Alibaba Group, [2]Harbin Institute of Technology,

**Correspondence:** yongjiang.jy@alibaba-inc.com {xcfeng, qinb}@ir.hit.edu.cn

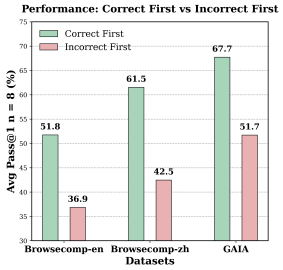
## Abstract

Large Language Model(LLM)-based agents have shown strong capabilities in web information seeking, with reinforcement learning (RL) becoming a key optimization paradigm. However, planning remains a bottleneck, as existing methods struggle with long-horizon strategies. Our analysis reveals a critical phenomenon—*plan anchor*—where the first reasoning step disproportionately impacts downstream behavior in long-horizon web reasoning tasks. Current RL algorithms, fail to account for this by uniformly distributing rewards across the trajectory. To address this, we propose Anchor-GRPO, a two-stage RL framework that decouples planning and execution. In Stage 1, the agent optimizes its first-step planning using fine-grained rubrics derived from self-play experiences and human calibration. In Stage 2, execution is aligned with the initial plan through sparse rewards, ensuring stable and efficient tool usage. We evaluate Anchor-GRPO on four benchmarks: BrowseComp, BrowseComp-Zh, GAIA, and XBench-DeepSearch. Across models from 3B to 30B, Anchor-GRPO outperforms baseline GRPO and First-step GRPO, improving task success and tool efficiency. Notably, WebAnchor-30B achieves 46.0% pass@1 on BrowseComp and 76.4% on GAIA. Anchor-GRPO also demonstrates strong scalability, getting higher accuracy as model size and context length increase.

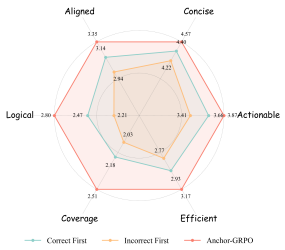
## 1 Introduction

Reinforcement Learning (RL) have significantly enhanced the capabilities of autonomous, tool-augmented agents built on large language models (LLMs) for web information seeking (Jin et al., 2025; Li et al., 2025a; Zhang et al., 2025a). These agents often referred to as deep research (OpenAI, 2025; Grok Team, 2025; Team et al., 2025), go beyond static retrieval and instead iteratively formulate queries, invoke external tools, and collect evidence from diverse online sources to answer complex questions.


(a) The plan anchor phenomenon, where the first-step decision disproportionately affects the trajectory's success.




(b) Impact of the first step on downstream task accuracy.




(c) Plan rubrics visualized through a radar chart.

Figure 1: Illustrating the plan anchor phenomenon, the critical role of the first step in task accuracy, and the use of plan rubrics to guide optimization.

Despite recent advancements, deep research agents still struggle with long-horizon planning and maintaining strategy coherence (Erdogan et al., 2025; Qiao et al., 2025). Increasing tool usage or managing context alone does not ensure consistent multi-step reasoning; instead, accumulated errors often lead to performance degradation (Liu et al., 2025). While existing reinforcement learning (RL) methods—such as context management, reward shaping, and entropy-based optimization—mitigate certain aspects, there are still gaps to address planning instability and the lack of a comprehensive long-term strategy (Chung et al., 2025; Wu et al., 2025; Zhao et al., 2025; Dong et al., 2025a).



*Findings of the Association for Computational Linguistics: ACL 2026*, pages 41368–41380
July 2-7, 2026 

A crucial insight is that the ability to maintain coherent long-horizon behavior hinges on early decisions (Su et al., 2025). Recent work highlights that not all reasoning steps are equally important (Bogdan et al., 2025). In particular, the first planning step often acts as a structural anchor—shaping exploration, tool usage, and evidence integration (Dong et al., 2025b). Initial missteps can trigger cascading failures and destabilize the entire trajectory (Sui et al., 2025; Sinha et al., 2025). Motivated by this, we identify a critical phenomenon in long-horizon web agents, which we term the ***plan anchor***, as shown in Fig. 1a. Our motivation experiments (Fig. 1b) demonstrate substantial performance drops in Avg Pass@1 across BC-ZH, BC-EN, and GAIA benchmarks due to incorrect first steps—by 28.7%, 30.9%, and 23.6%, respectively. These results underscore the importance of designing reward schemes that explicitly prioritize the impact of the first step.

To address this, we introduce ***Anchor-GRPO***, a two-stage RL framework that separates planning and execution. The first stage focuses on optimizing the initial planning step. In this stage, we develop the ***Plan Rubrics Learner***, by analyzing both successful and failed experiences, the learner adaptively refines essential planning criteria, such as task decomposition, goal alignment, and tool selection. The final rubrics assign higher scores to correct plans, as shown in Figure 1c, confirming the learner's ability to capture important planning capabilities. These refined rubrics are incorporated as reward signals in the RL process, guiding the agent to improve its planning capabilities. In Stage 2, the focus shifts to execution. Sparse rewards align execution with the initial plan, ensuring stability throughout the reasoning process and addressing long-horizon credit assignment challenges. By combining the power of plan rubrics and execution alignment, ***Anchor-GRPO*** enables agents to *plan first, then act*, resulting in a more reliable approach which can maintain consistency over long reasoning trajectories.

Our contributions are threefold:

- **Plan Anchor Phenomenon in Long-Horizon Web Reasoning:** We identify the critical phenomenon of ***plan anchor***, where a first-step decision disproportionately impacts the success of the entire trajectory, highlighting the importance of the initial planning step in long-horizon web reasoning.
- **Experience-based Plan Rubrics Learner:** We propose the Plan Rubrics Learner framework, which adaptively learns key dimensions of effective long-horizon plans from self-play experiences. The learnt rubrics will guide the agent in optimizing the first planning step for improved reasoning and task success.
- **Anchor-GRPO and Its Superior Performance:** We introduce Anchor-GRPO, a two-stage RL framework that enhances task success and long-horizon reasoning, outperforming existing methods on several challenging benchmarks. It also demonstrates scalability across model sizes and context lengths, making it suitable for more complex tasks.

## 2 Preliminary

**Agentic Web Reasoning.** We follow the standard ReAct-style agentic workflow (Yao et al., 2023), where an LLM-based agent interleaves *Thought*, *Action*, and *Observation*. At step $t$, the agent reads the trajectory history $\mathcal{H}_{t-1}$ and produces a reasoning trace $\tau_t$ and an executable action $a_t$, after which the environment returns an observation $o_t$. A $T$-step rollout is defined as:

$$\mathcal{H}_T = (q, \tau_1, a_1, o_1, \ldots, \tau_T, a_T, o_T, \tau_{T+1}, a_{T+1}),$$

where $q$ is the task query and $a_{T+1}$ denotes the final answer. At each step, the policy model samples:

$$(\tau_t, a_t) \sim \pi_\theta(\cdot \mid \mathcal{H}_{t-1}).$$

**Tool Design.** Following standard web agent designs (Li et al., 2025a; Gao et al., 2025), we define the action space with two tools for web exploration:

- **Search**: This tool issues top-$k$ web search queries to retrieve relevant snippets and URLs. It accepts natural language inputs and returns structured results from the Google Search API, including titles, snippets, and hyperlinks.
- **Visit**: Given a specific URL, the agent can browse the full page content and extract factual evidence. We use a language model browser to simulate document-level reading and structured content extraction.

Each action $a_t$ is thus instantiated as either `Search(query)` or `Visit(url)`. These tools enable multi-hop evidence gathering and decision-making under long-horizon uncertainty.

## 3 Methodology

In this section, we introduce a two-stage training framework Anchor-GRPO to optimize long-horizon reasoning in web agents. In the first stage, a ***Plan Rubrics Learner*** is used to derive effective planning criteria from past experiences, which are used to optimize the agent's initial planning with a dense reward. In the second stage, the agent's execution is optimized based on sparse rewards, with both stages jointly trained to align planning and execution.

### 3.1 Anchor-GRPO Overview

Anchor-GRPO is a two-stage reinforcement learning framework that integrates planning and execution within a single policy model. In the first stage, the model acts as a *Planner* ($\pi_{\text{planner}}$), generating an initial plan from the user query, optimized with dense rewards to improve task decomposition and planning quality. In the second stage, the model functions as an *Executor* ($\pi_{\text{executor}}$), executing the plan via tool interactions and receiving sparse rewards based on task success. Although both stages share the same model parameters, their training is decoupled through masked credit assignment: dense signals guide planning updates, while execution is refined using sparse feedback. This phased optimization aligns planning and execution without requiring separate models. We detail the reward design and masking-based credit assignment mechanism in the following sections.

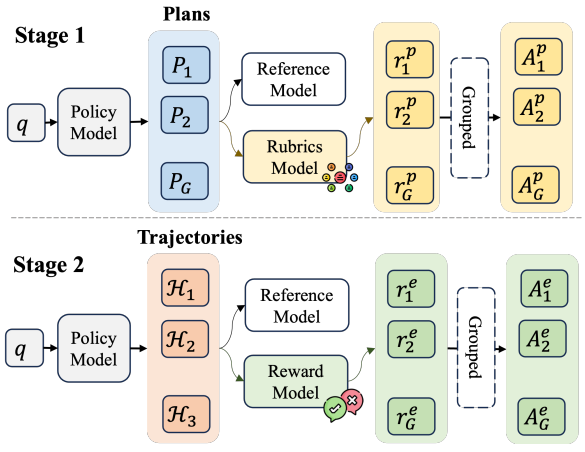



Figure 2: Anchor-GRPO framework. Stage 1 optimizes the initial plan using the Rubrics Model, providing dense rewards. Stage 2 refines the trajectory with sparse rewards, ensuring alignment with the plan.

### 3.2 Stage 1: Anchor Plan Optimization

The first stage improves the agent's initial planning by using Plan Rubrics learned from past experiences to define and reward good plans. This addresses the challenge of optimizing the first step in RL, ensuring each trajectory starts with a high-quality strategy and enhancing long-horizon performance.

#### 3.2.1 Plan Rubrics Learner

The Plan Rubrics Learner (Figure 3) distills planning principles from a large corpus of agent trajectories collected during web-agent training and evaluation. Through iterative refinement which guided by an LLM and human-in-the-loop feedback, it learns structured rubrics across key dimensions, each with fine-grained criteria. The resulting rubrics align with human judgments and reliably distinguish between correct and incorrect plans, enabling effective reward shaping for downstream planning.

**Insight Extraction** We extract planning insights from a diverse set of task trajectories collected during prior agent interactions, spanning both successful executions and failure cases, using three LLM-based functions: $\mathcal{F}_{\text{success}}$ and $\mathcal{F}_{\text{fail}}$ analyze correct and incorrect trajectories to identify effective heuristics and failure modes, while $\mathcal{F}_{\text{paired}}$ compares paired trajectories to infer decision boundaries. Each function outputs a tuple $s_i = (q_i, p_i, \text{insight}_i)$, where $q_i$ is the task query, $p_i$ the initial plan, and $\text{insight}_i$ the derived principle. We get the set of insights $S = \{s_1, s_2, \ldots, s_n\}$, defines a manifold of high-quality plans, separating successful from flawed strategies, and drives rubric refinement.

**Rubrics Optimization** We begin with an initial set of heuristic rubrics defined over $m$ planning dimensions $\{d_1, \ldots, d_m\}$ (e.g., task decomposition, goal alignment, tool selection). These rubrics are iteratively refined through alternating LLM-driven updates and human feedback.

At each iteration $t$, we sample a balanced batch $\mathcal{B}_t = \{\mathcal{B}_{\text{success}}, \mathcal{B}_{\text{failure}}, \mathcal{B}_{\text{paired}}\}$ from the insight set $S$, and update the rubrics using an LLM-based updater $\mathcal{F}_{\text{Update}}$:

$$\mathcal{R}_{t+1} = \mathcal{F}_{\text{Update}}(\mathcal{R}_t, \mathcal{B}_t),$$

where $r_t$ denotes the rubrics at step $t$.

After each epoch, we evaluate the rubrics on two criteria: (1) alignment with human judgments,

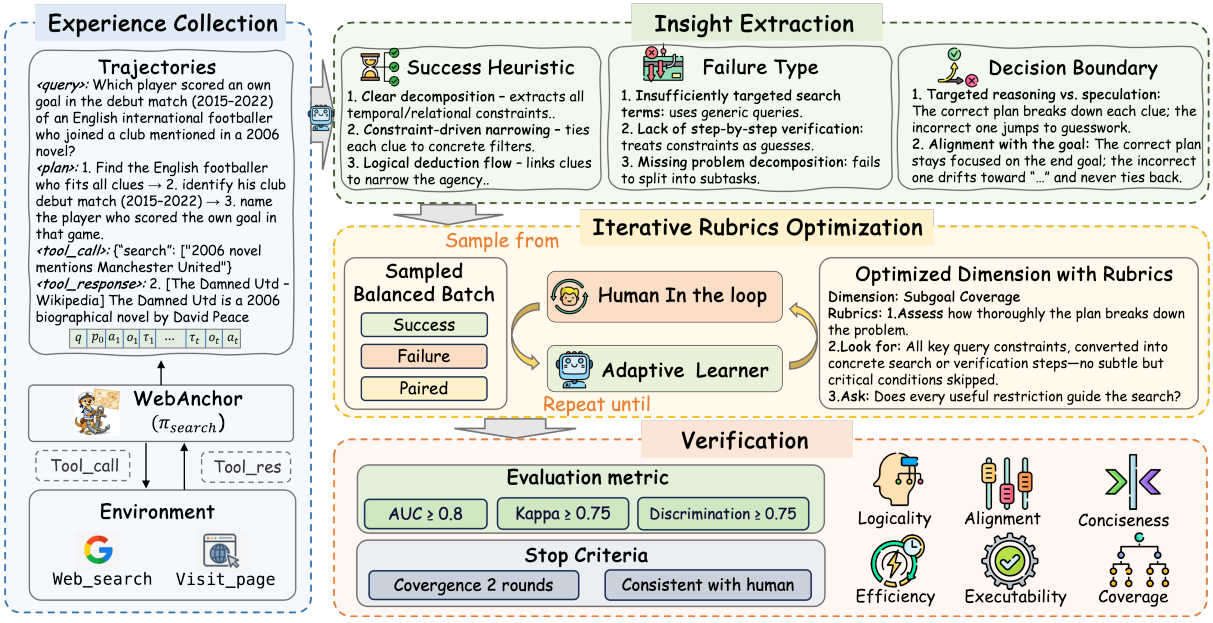



Figure 3: Overview of the Plan Rubrics Learner and verification process, illustrating how WebAnchor collects experiences, extracts insights, iteratively optimizes rubrics, and verifies plan quality with human feedback.

and (2) ability to discriminate between correct and incorrect plans via learned decision boundaries. When needed, human annotators refine ambiguous or erroneous rubric items to guide the next update cycle. The process continues until convergence criteria are met, yielding a robust rubric set that effectively shapes planning rewards.

### 3.2.2 Stage 1: Anchor Plan Optimization with Plan Rubrics

Stage 1 optimizes the *Anchor Plan*, where the agent's initial planning step decomposes the task, sets subtask goals, and selects tools. This stage uses a dense reward from the learned Plan Rubrics to improve first-step planning quality.

To focus credit assignment on the initial decision, we mask all actions after the first planning step during policy updates. Only the logits for the *Anchor Plan* are updated, while subsequent steps receive zero gradient.

**Plan Rubrics Reward Definition** Let $p$ denote the generated Anchor Plan for a given query. The Plan Rubrics define a normalized scoring function $\mathcal{R}(p) \in [0, 1]$, which evaluates $p$ across $m$ planning dimensions $\{d_1, \ldots, d_m\}$:

$$\mathcal{R}^{\text{plan}} = \mathcal{R}(p) = \frac{1}{Z} \sum_{j=1}^{m} \phi_j(p), \qquad (1)$$

where:

- $\phi_j(p) = \text{Judge}_{\text{LLM}}(p, d_j) \in [0, s_j^{\max}]$ is the LLM-based score for dimension $d_j$,
- $s_j^{\max}$ is the maximum achievable score for $d_j$,
- $Z = \sum_{j=1}^{m} s_j^{\max}$ is a normalization constant.

This reward is used as the immediate return for the first planning step in Stage 1, providing a dense, interpretable signal that aligns with experience-based planning quality.

**Objective Function for Plan Optimization** We optimize the *Anchor Plan* using a modified GRPO objective that operates only on initial plans. We sample a group of $G$ candidate plans $\{p_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}(\mathcal{P} \mid q)$ and maximize:

$$\mathcal{J}_{\text{plan}}(\theta) = E_{(q,a)\sim\mathcal{D},\{p_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\mathcal{P}|q)} \Bigg[ \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|p_i|} \sum_{j=1}^{|p_i|} \min \Big( r_{i,j}(\theta) \hat{A}_{i,j}^{\text{plan}}, \text{clip}(r_{i,j}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}) \hat{A}_{i,j}^{\text{plan}} \Big) \Bigg], \qquad (2)$$

where $r_{i,j}(\theta) = \frac{\pi_\theta(p_{i,j}|q,p_{i,<j})}{\pi_{\theta_{\text{old}}}(p_{i,j}|q,p_{i,<j})}$ is the importance sampling ratio at token $j$ of plan $i$, $\hat{A}_{i,j}^{\text{plan}}$ is the advantage estimate derived from the rubric $\mathcal{R}_i^{\text{plan}}$ via group normalization, and $\epsilon_{\text{low}}$, $\epsilon_{\text{high}}$ control the clipping range to stabilize policy updates.

**LLM Masking for First-Step Update** In this approach, the update for the planning policy $\pi_{\text{plan}}$ is confined to the first step of the trajectory. The LLM's outputs for subsequent steps are masked, ensuring that only the first step's plan influences the update. This isolates the planner's optimization to the initial decision, preventing interference from later trajectory interactions or the executor policy.

### 3.3 Stage 2: Trajectory Level Executor Optimization

**Executor Reward** In the second stage, the executor is trained to follow the optimized plan using a sparse task-completion reward. The reward is assigned only at the end of the episode based on whether the final answer exactly matches the ground truth:

$$\mathcal{R}^{\text{exec}} = \begin{cases} 1 & \text{if } \texttt{ExactMatch}(\text{Answer}, \text{GT}), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

All intermediate steps receive zero reward ($r_t^{\text{exec}} = 0$ for $t < T$), with the full reward $r^{\text{exec}}$ assigned only at termination ($t = T$), encouraging the executor to follow the plan and achieve task success.

**Objective for Execution** In Stage 2, we optimize the executor policy at the trajectory level. We sample a group of $G$ rollouts $\{\mathcal{H}^{(i)}\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}(\mathcal{H} \mid q)$ and maximize:

$$\mathcal{J}_{\text{exec}}(\theta) = E_{(q,a)\sim\mathcal{D},\{H^{(i)}\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\mathcal{H}|q)} \Bigg[ \frac{1}{G}\sum_{i=1}^{G} \frac{1}{|\mathcal{H}^{(i)}|} \sum_{j=1}^{|\mathcal{H}^{(i)}|} \min\Big( r_{i,j}(\theta)\hat{A}_{i,j}^{\text{exec}}, \text{clip}(r_{i,j}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_{i,j}^{\text{exec}} \Big) \Bigg], \tag{4}$$

where $r_{i,j}(\theta)$, $\epsilon_{\text{low}}$, and $\epsilon_{\text{high}}$ are defined identically to those in Equation (2), while $\hat{A}_{i,j}^{\text{exec}}$ is the advantage estimate derived from the trajectory-level execution reward $\mathcal{R}^{\text{exec}}$ via group normalization.

### 3.4 Plan Rubrics Evaluation and Validation

Rubrics are validated at the end of each optimization cycle against manual plan outcomes using two metrics: AUC (measuring ranking quality) and Cohen's $\kappa$ (measuring agreement). We require

$$\text{AUC} \geq 0.8, \quad \kappa \geq 0.75.$$

If either threshold is not met, the rubrics will be revised. This iterative validation–revision loop continues until convergence, ensuring the rubrics remain aligned with task success criteria.

## 4 Experiments

In this section, we present a series of experiments designed to evaluate the effectiveness and scalability of Anchor-GRPO, addressing three critical questions:

**1. Performance of Anchor-GRPO:** Does WebAnchor outperform strong external agents, including proprietary models (e.g., OpenAI, DeepSeek-V3.1) and open-source models (e.g., R1-Searcher (Song et al., 2025), WebSailor), as well as internal baselines such as GRPO and First-step GRPO, across challenging agentic benchmarks?

**2. Plan Quality and Downstream Impact:** How does Anchor-GRPO enhance the quality of initial plans, particularly in terms of Subgoal Coverage, Goal Alignment, and Tool Efficiency? More importantly, how does this improvement in planning quality translate to better downstream execution and task success?

**3. Scalability Potential:** Is the two-stage design of Anchor-GRPO well-suited for future scaling? We examine whether performance consistently improves as model sizes increase (ranging from 3B to 30B) and context lengths grow (from 16k to 64k), suggesting strong potential for continued improvement as model capacity and complexity advance.

**Ablation Studies:** We conduct ablation studies that evaluate the necessity of core elements—such as two-stage optimization, rubric-based reward shaping, and first-step planning. Additionally, we analyze how the quality of the Anchor Plan impacts tool usage efficiency and overall task performance.

### 4.1 Experimental Setup

#### 4.1.1 Benchmarks and Metrics

**Benchmarks** We evaluate our method on four challenging benchmarks for web-based information-seeking tasks: BrowseComp_en (Wei et al., 2025), BrowseComp_zh (Zhou et al., 2025), XBench-DeepSearch (Xbench Team, 2025), and GAIA (Mialon et al., 2023).

**Metrics** We evaluate using Pass@1 and Pass@3, which measure the success rates of finding the correct answer in the first and top-three rollouts, re-

| Model | RL Algo | GAIA | | BrowseComp | | BrowseComp-ZH | | Xbench | |
|---|---|---|---|---|---|---|---|---|---|
| | | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 |
| *Advanced Models* | | | | | | | | | |
| OpenAI-o3† | - | 70.5 | - | 50.9 | - | 58.1 | - | 66.7 | - |
| Claude-4-Sonnet† | - | 68.3 | - | 12.2 | - | 29.1 | - | 64.6 | - |
| Kimi-K2† | - | 57.7 | - | 14.1 | - | 28.8 | - | 50.0 | - |
| DeepSeek-V3.1† | - | 63.1 | - | 30.0 | - | 49.2 | - | 71.0 | - |
| *Open-source Agents* $\leq$ *32B* | | | | | | | | | |
| R1-Searcher-7B | - | 20.4 | - | 0.4 | - | 0.6 | - | 4.0 | - |
| WebThinker-RL | - | 48.5 | - | 2.8 | - | 7.3 | - | 24.0 | - |
| WebDancer-QwQ | - | 51.5 | - | 3.8 | - | 18.0 | - | 39.0 | - |
| WebSailor-7B | - | 37.9 | - | 6.7 | - | 14.2 | - | 34.3 | - |
| WebSailor-32B | - | 53.2 | - | 10.5 | - | 25.5 | - | 53.3 | - |
| **WebAnchor-3B** | GRPO | 27.2 | 43.6 | 5.1 | 8.7 | 12.3 | 19.2 | 31.6 | 55.0 |
| | First-step GRPO | **30.1** | **44.6** | 6.9 | 10.2 | 12.9 | 21.5 | 34.7 | 55.0 |
| | **Anchor-GRPO** | 28.3 | 43.5 | **7.1** | **11.5** | **13.1** | **25.0** | **37.7** | **56.0** |
| **WebAnchor-7B** | GRPO | 33.0 | 44.7 | 6.3 | 11.7 | 17.5 | 31.5 | 40.7 | 56.0 |
| | First-step GRPO | **38.5** | **55.1** | 9.3 | 16.3 | 16.4 | 31.2 | 37.7 | 52.0 |
| | **Anchor-GRPO** | 37.8 | 51.5 | **11.6** | **20.7** | **23.8** | **36.7** | **45.0** | **66.0** |
| **WebAnchor-30B** | GRPO | 75.8 | **87.6** | 44.0 | 65.0 | 46.2 | 61.4 | 71.0 | 84.0 |
| | First-step GRPO | 74.8 | 87.4 | 43.3 | 66.0 | 47.9 | 63.2 | 72.0 | **88.0** |
| | **Anchor-GRPO** | **76.4** | 86.4 | **46.0** | **67.5** | **48.8** | **65.1** | **75.1** | 85.0 |

Table 1: Performance comparison of the proposed Anchor-GRPO method across different model sizes and RL algorithms, evaluated on multiple benchmarks (GAIA, BrowseComp, BrowseComp-ZH, Xbench).

spectively. Pass@1 is averaged over three runs for stability. We use Qwen-2.5-72B as scoring model.

#### 4.1.2 Baselines

We conduct experiments on models ranging from 3B to 30B: including WebSailor-3B [1] , WebSailor-7B [2] , and Tongyi-DR-30B [3] , which are fine-tuned on synthetic information-seeking data for multi-turn tool use and serve as the base policies. We compare three RL settings, including our method: **1. GRPO**, which applies GRPO over the entire trajectory using exact match reward for all planning and execution steps; and **2. First-step GRPO**, a two-stage approach that first optimizes the initial planning step and then optimizes the entire trajectory. Both stages use sparse (0/1) rewards based on final answer exact match. Our method, **3. Anchor-GRPO**, also uses a two-stage process but leverages Plan Rubrics–derived dense rewards for initial plan optimization, followed by sparse task-completion rewards during execution.

**Training Setup** We use 1,000 high-quality examples from an in-house wiki corpus, filtered by task difficulty. The training process takes place in a virtual wiki environment, replaces the real web to ensure faster request speeds and improved stability. Our experiments are conducted on 64 GPUs, with batch size of 32 and rollout num of 8.

[1] https://huggingface.co/Alibaba-NLP/WebSailor-3B

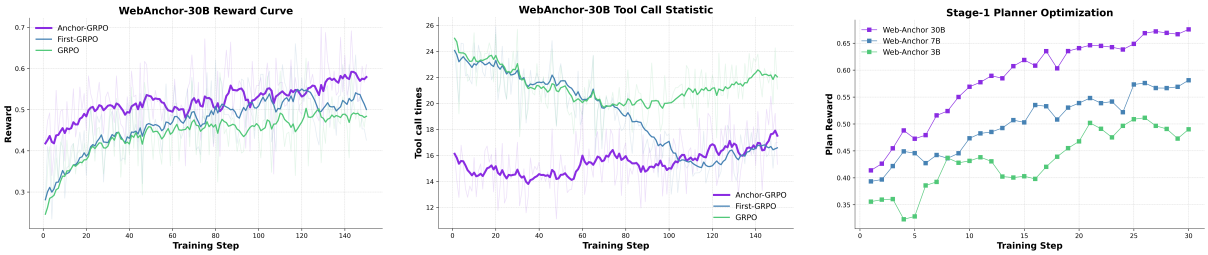
[2] https://huggingface.co/Alibaba-NLP/WebSailor-7B

[3] https://huggingface.co/Alibaba-NLP/Tongyi-DeepResearch-30B-A3B

### 4.2 Main Results

**Anchor-GRPO Consistently Outperforms Baselines and Achieves SOTA Performance Across Different Model Sizes** Anchor-GRPO outperforms both baseline GRPO and other models, achieving SOTA performance in Pass@1 and Pass@3 across multiple benchmarks. For example, WebAnchor-30B achieves **46.0% Pass@1** on BrowseComp, surpassing baseline GRPO (42.0%) and First-step GRPO (41.3%). In GAIA, WebAnchor-30B achieves **76.4% Pass@1**, outperforming WebSailor-32B (53.2%) and OpenAI-o3 (70.5%). These results highlight Anchor-GRPO's ability to improve task completion rates and surpass existing methods in long-horizon reasoning.

**First-GRPO Demonstrates the Effectiveness of Two-Stage Training, While Anchor-GRPO Further Optimizes with Plan Rubrics Reward** First-step GRPO demonstrates the benefits of two-stage training by optimizing the first step independently. Anchor-GRPO further enhances performance by integrating a **Plan Rubrics Reward**




(a) Training dynamics comparison across GRPO, First GRPO, and Anchor-GRPO.

(b) Tool-call usage comparison across GRPO, First GRPO, and Anchor-GRPO.

(c) Rubrics reward scaling of Anchor-GRPO across model sizes (3B–30B).

Figure 4: Performance and behavior analysis of Anchor-GRPO versus baselines, including training convergence, tool efficiency, and reward robustness across model scales.

in Stage 1. This reward significantly improves the first-step planning, as shown by the **2.7% improvement in Pass@1** for WebAnchor-30B over First-step GRPO at BC, underscoring the advantage of planning optimization through structured rubrics.

**Anchor-GRPO Exhibits Strong Scalability Across Different Model Sizes** Anchor-GRPO shows excellent scalability, with performance improving from 3B to 30B models. WebAnchor-30B achieves **76.4% Pass@1** on GAIA, significantly outpacing WebAnchor-7B (37.8% Pass@1). In XBench, WebAnchor-30B reaches **75.1% Pass@1**, showing robust performance as model size increase, proving its potential for complex tasks.

### 4.3 Training Dynamics

**Reward Dynamics** The WebAnchor-30B training curve shows in 4a that Anchor-GRPO consistently outperforms both First-GRPO and GRPO, with a steady increase in rewards. This improvement is driven by Stage 1's Plan optimization, which strengthens the agent's first-step planning. By decoupling planning and execution, Anchor-GRPO provides a stable foundation for higher performance throughout training.

**Tool Calling Dynamics** As shown in 4b, Anchor-GRPO exhibits more efficient and stable tool usage compared to both GRPO and First-GRPO. While First-GRPO reduces tool calls, it limits task success by restricting exploration. In contrast, Anchor-GRPO optimizes tool usage, balancing efficiency with performance, resulting in better task completion.

### 4.4 Ablation Studies

**Two-Stage Training Strategy** We compare three settings: (i) standard GRPO (baseline), (ii) Stage-1-only, and (iii) full two-stage GRPO. While Stage-1-only yields the highest Alignment (67.2) and Efficiency (65.4), its Pass@1 (42.0) lags behind the full method. The two-stage approach achieves the best task success (Pass@1: **46.0**), demonstrating that joint optimization enables the agent to adaptively refine execution based on high-quality plans.

**First-Step vs. Other-Step Update** We ablate which step is updated during GRPO: first, last, or a random intermediate step. First-Step GRPO achieves the highest Pass@1 (43.3) and Alignment (58.2), significantly outperforming Random Step (33.0) and Last Step (36.1). This confirms that optimizing the initial planning decision provides a critical anchor for long-horizon task completion.

**Reward Design for Planning** We evaluate three reward schemes for the planner: (i) sparse 0–1 terminal reward, (ii) Naive Plan Reward, and (iii) our dense rubric-based reward. The dense rubric reward leads to the strongest performance (Pass@1: **46.0**), substantially improving over Naive (44.2) and terminal (43.3) rewards, highlighting the value of structured, multi-dimensional feedback in shaping effective plans.

### 4.5 Scaling of Anchor-GRPO

**Context length scaling** We evaluate Anchor-GRPO on BrowseComp_EN with context lengths of 32k, 48k, and 64k. Results shows in 5 consistent performance gains as context length increases, demonstrating effective scaling. This suggests that Anchor-GRPO will further benefit from models with larger context windows or greater capacity.

**Plan rubrics reward scaling** As shown in 4c, WebAnchor models (3B, 7B, and 30B) all successfully converge on the plan rubrics reward, with reward values increasing monotonically with model size. This demonstrates strong scaling behavior of

| Ablation Settings | Pass@1 | Pass@3 | Coverage | Alignment | Efficiency | Avg. Tool Calls |
|---|---|---|---|---|---|---|
| *Two-Stage Training Strategy* | | | | | | |
| Full GRPO (baseline) | 44.0 | 65.0 | 36.0 | 61.2 | 53.8 | 15.5 |
| Only Planner Updated (Stage-1) | 42.0 | 57.0 | 48.2 | **67.2** | **65.4** | 14.9 |
| **Two-Stage (Planner + Executor)** | **46.0** | **67.5** | **50.2** | 67.0 | 63.4 | 17.3 |
| *First-Step vs. Other-Step Update* | | | | | | |
| Random Step GRPO | 33.0 | 38.5 | 39.0 | 43.6 | 46.6 | 16.2 |
| Last Step GRPO | 36.1 | 40.8 | 33.4 | 41.4 | 42.8 | 13.0 |
| **First-Step GRPO** | **43.3** | **66.0** | **44.4** | **58.2** | **56.6** | 14.6 |
| *Reward Design* | | | | | | |
| 0-1 Terminal Reward | 43.3 | 66.0 | 39.8 | 43.4 | 47.8 | 15.7 |
| Naive Plan Reward | 44.2 | 63.4 | 44.6 | 52.2 | 51.4 | 19.2 |
| **Planner Dense Reward** | **46.0** | **67.5** | **50.2** | **67.0** | **63.4** | 17.3 |

Table 2: Ablation studies on Browsecomp_en across three design dimensions. We compare how different settings affect performance, plan quality and tool call efficiency.

the learned rubrics with respect to model capacity.

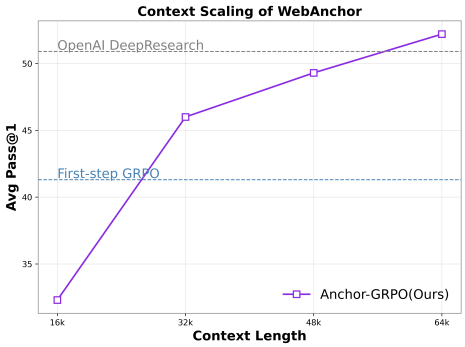



Figure 5: Scaling of Anchor-GRPO

## 5 Related Work

**Long Horizon Web Reasoning** Recent Deep Research (DR) agents tackle long-horizon web reasoning through multi-step planning, iterative refinement, and web-scale evidence synthesis. Moving beyond standard RAG, works like WebThinker (Li et al., 2025b) and WebResearcher (Qiao et al., 2025) integrate Large Reasoning Models (LRMs) for deep thinking and self-correction. To manage unstructured evidence, structural frameworks such as WebWeaver (Li et al., 2025c) employ adaptive hierarchies to preserve coherence. These advances are benchmarked by DeepResearch Bench (Du et al., 2025) and extended to multimodal settings (Geng et al., 2025)), forming a robust foundation for autonomous scientific research.

**Agentic Reinforcement Learning** Agentic RL has emerged as a key paradigm, transforming LLMs from passive sequence generators to autonomous agents capable of environmental interaction and multi-step decision-making. Foundational surveys have formalized this transition, highlighting how RL optimizes both internal reasoning and external actions of agents (Zhang et al., 2025a). Multi-turn training frameworks like Agent-R1 (Cheng et al., 2025) and AgentGym-RL (Xi et al., 2025) enhance long-horizon performance and tool-use capabilities. Researchers are also addressing challenges like sparse feedback and robust tool integration through novel reward structures, as seen in VerlTool (Jiang et al., 2025). These advancements, supported by meta-thinking frameworks such as ReMA (Wan et al., 2025) and verifiable reasoning models like RLVMR (Zhang et al., 2025b), create a comprehensive ecosystem where RL powers tool-augmented AI systems.

## 6 Conclusion

We present Anchor-GRPO, a two-stage reinforcement learning framework that decouples planning and execution to address the unique challenges of long-horizon web reasoning. By introducing Plan Rubrics Learner, structured criteria distilled from agent experiences, we enable dense and interpretable reward shaping that significantly improves plan quality. Our ablation studies confirm that (1) optimizing the first planning step acts as a critical anchor for downstream success, (2) joint planner-executor training yields superior task accuracy over planner-only optimization, and (3) rubric-based dense rewards are essential for effective policy learning. Evaluated on complex web research tasks, Anchor-GRPO achieves state-of-the-art performance, demonstrating that principled planning grounded in explicit reasoning standards is key to building stable agents.

## Limitations

In this work, we have focused on applying the method to web agents and related tasks. However, we believe the plan anchor phenomenon may also be relevant in other domains, and we look forward to exploring the potential of this method in those areas. WebAnchor still has significant room for improvement in the proposed plan rubrics, which could potentially lead to further performance gains. We also hope that future research will continue to highlight the importance of first-step planning and introduce new optimization techniques to enhance its effectiveness.

## Acknowledgments

Xiaocheng Feng, and Bing Qin are the co-corresponding authors of this work. We thank the anonymous reviewers for their insightful comments. This work was supported by the National Natural Science Foundation of China (NSFC) (grant 62522603, 62276078), the Key R&D Program of Heilongjiang via grant 2022ZX01A32, the Fundamental Research Funds for the Central Universities ( XNJKKGYDJ2024013 ) .

## References

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.

Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruiran Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. 2025. Agent-r1: Training powerful llm agents with end-to-end reinforcement learning. *Preprint*, arXiv:2511.14460.

Andy Chung, Yichi Zhang, Kaixiang Lin, Aditya Rawal, Qiaozi Gao, and Joyce Chai. 2025. Evaluating long-context reasoning in llm-based webagents. *arXiv preprint arXiv:2512.04307*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025a. Agentic reinforced policy optimization. *Preprint*, arXiv:2507.19849.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. 2025b. Emergent response planning in llms. *arXiv preprint arXiv:2502.06258*.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *Preprint*, arXiv:2508.07976.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, and 1 others. 2025. Webwatcher: Breaking new frontiers of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*.

Grok Team. 2025. Grok-3 deeper search.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, Tianyu Pang, and Wenhu Chen. 2025. Verltool: Towards holistic agentic reinforcement learning with tool use. *Preprint*, arXiv:2509.01055.

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

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

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

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, Jun Zhang, and Jingren Zhou. 2025c. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *Preprint*, arXiv:2509.13312.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, and 1 others. 2025. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*.

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

OpenAI. 2025. Deep research system card.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, and 1 others. 2025. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309*.

Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping. 2025. The illusion of diminishing returns: Measuring long horizon execution in llms. *arXiv preprint arXiv:2509.09677*.

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

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang, Huifeng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, and 1 others. 2025. Scaling agents via continual pre-training.

Yuan Sui, Yufei He, Tri Cao, Simeng Han, Yulin Chen, and Bryan Hooi. 2025. Meta-reasoner: Dynamic guidance for optimized inference-time reasoning in large language models. *arXiv preprint arXiv:2502.19918*.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. 2025. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *Preprint*, arXiv:2503.09501.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, and 1 others. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

Xbench Team. 2025. Xbench-deepsearch.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, and 4 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *Preprint*, arXiv:2509.08755.

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

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, and 6 others. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2025b. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *Preprint*, arXiv:2507.22844.

Yida Zhao, Kuan Li, Xixi Wu, Liwen Zhang, Dingchu Zhang, Baixuan Li, Maojia Song, Zhuo Chen, Chenxi Wang, Xinyu Wang, and 1 others. 2025. Repurposing synthetic data for fine-grained search agent supervision. *arXiv preprint arXiv:2510.24694*.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

# A Appendix

## A.1 Motivation Experiment Details

We use the Tongyi-deepresearch-A30 model to generate three rounds of rollouts for each dataset: Browsecomp-en, Browsecomp-zh, and GAIA. We select queries that are neither all wrong nor all correct. We separately select the correct first step and the incorrect first step. Then, we fix the first step and generate 8 rollouts. We calculate the average Pass@8 for the correct first step and the incorrect first step. We found significant dropouts in BC-ZH, BC-EN, and GAIA, with drops of 28.76%, 30.89%, and 23.63%, respectively. These results highlight the significant effect of the first step anchoring.

## A.2 Detailed Prompts

### A.2.1 Insight Extraction Prompt

**Single Insight Extraction Prompt** This prompt is used to extract insight from single successful or failed trajectory.

> You are an expert in evaluating reasoning plans.
> Query: question
> Plan (known to be correctness): plan
> Task: Analyze what makes this plan correctness. Focus on: - Specific behaviors that contribute to success (if correct) - Critical flaws that cause failure (if incorrect) - Concrete dimensions of planning quality
> Output only a concise bullet-point list of insights.

**Paired Insight Extraction Prompt** This prompt is used to extract insight from single successful or failed trajectory.

> You are an expert in evaluating reasoning plans for web-based information-seeking tasks.
> Below is a query and two plans: one that leads to the correct answer, and one that fails.
> Query: question
> Correct Plan: correct_plan
> Incorrect Plan: incorrect_plan
> Task: Analyze the key differences between these two plans. Specifically: 1. What does the correct plan do well that the incorrect plan misses? 2. Which dimensions of planning quality are most discriminative here? 3. Avoid vague statements; be concrete and grounded in the text.
> Output only a concise bullet-point list of insights (no intro/outro)

### A.2.2 Plan Rubrics Prompt

> You are tasked with evaluating the following plan for a web information seeking task. Please score the plan on the following dimensions using a scale from 0 to 5:
> - **0 = Very Poor**: Plan barely addresses the criteria or is largely incorrect. - **1 = Poor**: Plan meets only a small part of the criteria and is mostly ineffective. - **2 = Fair**: Plan is somewhat effective and meets basic aspects, though with flaws or inefficiencies. - **3 = Good**: Plan is reasonably effective and meets many of the criteria. - **4 = Very Good**: Plan is mostly effective and meets most criteria with minor issues. - **5 = Excellent**: Plan is highly effective, efficient, and well-structured.
> Plan and Query:
> - **Query**: <query> - **Plan**: <plan>
> Please rate the plan on the following dimensions, considering the **query** as the task goal and evaluating how well the plan addresses the task: 1. Goal Alignment What to assess: Whether the plan stays focused on what the user actually wants. Look for: A clear sense of the final answer—what kind of thing it is (a name, a number, a date) and how the query's details shape it. The best plans show how each clue helps zero in on that target. Ask yourself: Does this plan really understand what needs to be found—and why?

2. Subgoal Coverage What to assess: How thoroughly the plan breaks down the problem. Look for: All key constraints from the query (time, place, people, numbers, relationships) turned into concrete search or verification steps. Strong plans don't skip subtle but critical conditions. Ask yourself: Has every meaningful restriction been accounted for in a way that guides the search?
3. Tool Appropriateness What to assess: Whether the right sources are chosen for the job. Look for: Use of authoritative, relevant resources (e.g., official filings for financial data, academic databases for research), with awareness of how to access them. Bonus if it considers alternatives when a source might fail. Ask yourself: Are these the most trustworthy and efficient places to get this information?
4. Logical Ordering What to assess: The flow of reasoning from start to finish. Look for: A natural progression—starting broad or with high-signal clues, then narrowing down step by step. Each action should set up the next, not repeat or jump ahead. Ask yourself: Does the sequence feel like a smart, efficient path to the answer?
5. Actionability What to assess: Whether the plan can actually be carried out. Look for: Concrete, unambiguous instructions that a person (or agent) could follow without guessing. Vague phrases like "look it up" weaken this dimension. Ask yourself: Could someone execute this as written—or would they need to fill in gaps?
6. Clarity and Conciseness What to assess: How easy the plan is to read and follow. Look for: Clean structure, consistent language, and no unnecessary repetition. Good plans are brief but complete—nothing missing, nothing extra. Ask yourself: Is this plan easy to understand at a glance?
Output Format: Please output your evaluation in **JSON format** with the following structure: "Goal Alignment": "score": [SCORE], "comment": "[COMMENT]" , "Subgoal Coverage": "score": [SCORE], "comment": "[COMMENT]" , "Tool Appropriateness": "score": [SCORE], "comment": "[COMMENT]" , "Logical Ordering": "score": [SCORE], "comment": "[COMMENT]" , "Actionability": "score": [SCORE], "comment": "[COMMENT]" , "Clarity and Conciseness": "score": [SCORE], "comment": "[COMMENT]" , "total_score": [TOTAL SCORE], "overall_comment": "[OVERALL COMMENT]"
Where:
[SCORE] is a number from 0 to 5 based on the evaluation criteria.
[COMMENT] provides an explanation of the score.
[TOTAL SCORE] is the sum of the individual scores.
[OVERALL COMMENT] is a brief comment summarizing the overall quality of the plan.

### A.3 Pseudo code of Plan rubrics optimization

**Algorithm 1** Stage 1: Anchor Plan Optimization via Rubric-Guided Learning

**Require:** Insight set $S = \{s_i = (q_i, p_i, \text{insight}_i)\}_{i=1}^{n}$ from prior trajectories
**Require:** Initial rubrics $\mathcal{R}_0$ over planning dimensions $\{d_1, \ldots, d_m\}$
**Require:** LLM-based updater $\mathcal{F}_{\text{Update}}$, convergence criterion
Initialize rubrics: $\mathcal{R} \leftarrow \mathcal{R}_0$
**repeat**
    Sample balanced batch $\mathcal{B}_t = \{\mathcal{B}_{\text{success}}, \mathcal{B}_{\text{failure}}, \mathcal{B}_{\text{paired}}\} \subseteq S$
    Update rubrics: $\mathcal{R} \leftarrow \mathcal{F}_{\text{Update}}(\mathcal{R}, \mathcal{B}_t)$
    Evaluate $\mathcal{R}$ on:
(i) alignment with human judgments
(ii) discriminative power between correct/incorrect plans
    **if** human feedback needed **then**
        Refine ambiguous/erroneous rubric items via annotators
    **end if**
**until** convergence criteria met
**Ensure:** Final rubric set $\mathcal{R}$