CREAT: A Comprehensive Reward Bench for Lengthy and Complex Web Agent Trajectories

Anonymous ACL submission

Abstract

Recent advances in large language models (LLMs) have produced capable web agents, and evaluating their action trajectories is critical for post-training data selection and feedback-driven improvement. However, this assessment space remains under-explored, existing benchmarks emphasize short, simple tasks and primarily evaluate trajectory correctness. As agent capabilities grow and attention shifts to realistic, complex scenarios, modern web agents routinely engage in long-horizon reasoning over dozens of turns, which poses new challenges for evaluation. To better suit real-world evaluation needs, we present **CREAT**—**C**omprehensive **RE**ward bench for lengthy and complex web **A**gent **T**rajectories—a benchmark of complex queries that require high-order web browsing and long-horizon agentic reasoning. **CREAT** evaluates trajectories along five fine-grained dimensions that are crucial for web agents, going beyond correctness alone. Experiments on 10 mainstream LLMs reveal weak sensitivity to hallucinations and a limited ability to separate necessary exploration from redundant actions, providing insights about whether current LLMs can serve as reliable judges for comprehensive agent trajectory evaluation. [1]

1 Introduction

In recent years, the rapid advancement of Large Language Models (LLMs) (Grattafiori et al., 2024; DeepSeek-AI et al., 2025; OpenAI, 2025) and Multimodal Large Language Models (MLLMs) (Liu et al., 2023; OpenAI et al., 2024; Bai et al., 2025) have made it increasingly feasible to develop powerful web agents capable of interacting with web browser (Nakano et al., 2022; Ma et al., 2024; He et al., 2024; Xu et al., 2025). Through observing web pages and generating fine-grained, human-like operations such as clicking and typing, state-of-the-art agents already demonstrate remarkable progress and, in some cases, promising human-level behaviors on real-world tasks (Wei et al., 2025; Qi et al., 2025; Wang et al., 2025).

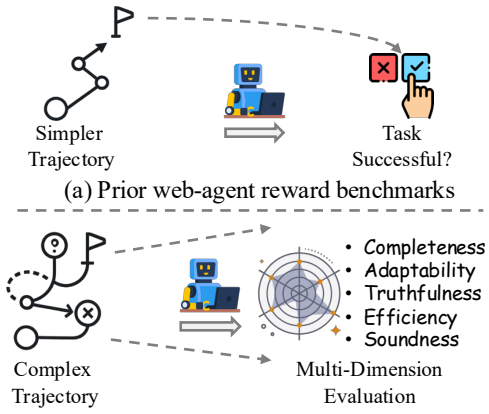


Figure 1: Comparison between (a) prior web-agent reward benchmarks, which largely emphasize simple tasks and correctness-only scoring, and (b) our CREAT benchmark, which targets long-horizon, complex trajectories and multi-dimensional evaluation.

To further enhance the performance of web agent, the assessment of agent trajectories (i.e. the reasoning and action sequence generated by agent) is crucial. High-quality assessments can (i) filter trajectories for post-training or preference optimization (Qi et al., 2025; Bai et al., 2024) and (ii) provide actionable feedback so agents can understand errors and correct them (Song et al., 2024; Zhai et al., 2024). Recent work shows that LLMs can serve as automatic judges of web-agent trajectories (Pan et al., 2024a; Murty et al., 2025); however, systematic studies of how well LLMs evaluate web agents, especially on long-horizon and complex tasks, remain limited. Several web agent reward benches (Lù et al., 2025) have emerged, but most of them center on relatively simple tasks that require minimal strategy, or they primarily assess

[1] Our data and code will be publicly released.

trajectory correctness (success/failure) as shwon in Figure 1(a). Meanwhile, modern web agents increasingly tackle realistic, complex tasks that demand long-horizon reasoning, sometimes spanning tens of turns (e.g., GAIA-style tasks (Mialon et al., 2023)). Such trajectories introduce new challenges for LLM-based evaluation: inputs are longer and noisier; and correctness alone is insufficient to guide agent evolution. Evaluations must therefore consider multiple dimensions beyond outcome accuracy. This raises a central question: **Can LLMs serve as reliable judges for comprehensively evaluating lengthy, complex web-agent trajectories?**

To investigate this problem, we introduce **CREAT** — a **C**omprehensive **RE**ward benchmark for lengthy and complex web **A**gent **T**rajectories — designed to assess LLM-based judges in this setting. To ensure task complexity, we select 95 questions that requires higher-order browser operation from GAIA (Mialon et al., 2023). To better elicit complex, hybrid tool-use behavior, we build a unified environment that supports both GUI operations (click, type, scroll, etc.) and a suite of MCP (Model Context Protocol) compatible tools (e.g., download, add-note). Using this environment, we employ strong commercial models to generate a large pool of candidate trajectories and then engage PhD-level annotators to curate representative examples containing both good and bad behaviors.

To enable fine-grained analysis of LLM judges, we evaluate the chosen trajectories along **five dimensions**: **(1) Completeness:** Extent of correct progress achieved before the first uncorrected deviation from the intended goal occurs. **(2) Adaptability:** Ability to avoid stagnation or persistent errors and adjust strategy to resume effective progress. **(3) Truthfulness:** Accuracy of facts and reasoning, and the capacity to correct false or flawed outputs. **(4) Efficiency:** Proportion of redundant or incorrect ones. **(5) Soundness:** Degree to which reasoning process are well-supported by sufficient and relevant evidence. After stringent manual annotation on these fine-grained evaluation dimensions and quality control, we obtain 190 trajectories for CREAT. We then benchmark 10 mainstream models as judges and conduct a comprehensive analysis, including the effect of different trajectory complexities as well as potential biases. Our analysis reveals the limitations of LLMs in performing holistic evaluations, their unawareness of hallucinations, and their insufficient ability to distinguish exploratory steps from redundant ones.

Our contributions can be summarized as follows:

- We develop CREAT, a benchmark designed to investigate whether current LLMs can serve as reliable judges for comprehensive and lengthy agent trajectory evaluation.
- CREAT comprises 95 complex problems that demand high-order web-browsing operations and long-horizon agentic reasoning, together with 190 high-quality trajectories. It further introduces five fine-grained evaluation dimensions aligned with practical error types, with well-defined scoring criteria on a four-point scale (1–4).
- We conduct experiments on 10 mainstream models and perform an in-depth analysis, uncovering their unawareness of hallucinations and their limited ability to differentiate between exploratory and redundant steps.

2 Environment

Our objective is to collect trajectories of Web Agents performing tasks that are more complex, requiring stronger reasoning, planning capabilities, and the use of auxiliary tools. To achieve this, while retaining the key characteristics of environments used in most previous web agent benchmarks, we introduce certain modifications and design a unified environment to help agents enhance their capabilities and enable the collection of trajectories that meet our requirements. In addition, we execute tasks on live, real-world web pages to ensure the authenticity of the trajectories.

2.1 Agent

We formulate the web agent as a Markov Decision Process (MDP) $\langle S, A, T, P \rangle$. The state space S represents the state of the current browser environment and the internal state of the agent. The action space A consists of all valid tool calls to the given tool servers. The transition function $T : S \times A \rightarrow S$ is deterministic. The plan P specifies the overall execution strategy and the main steps for accomplishing the task.

At each time step t, the agent is provided with an observation of the current state $s_t \in S$, which includes o_t and h_t, as well as the plan P. The agent must generate the next action $a_t \in A$, and subsequently receives a response r_t from the MCP server. Here, o_t contains the current web page's Accessibility Tree and its URL, while h_t denotes the history of interactions with tool servers, i.e.,

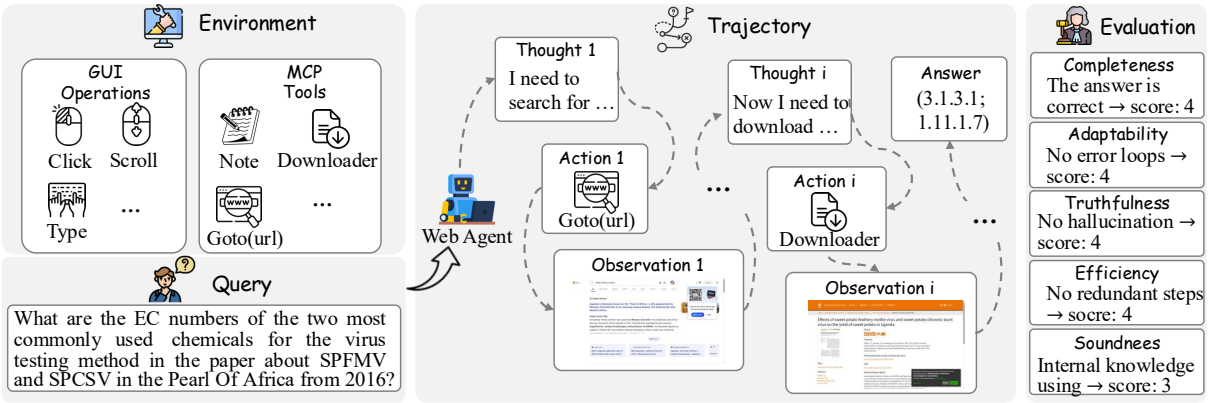


Figure 2: Overview of CREAT. The left part illustrates the unified environment we implemented, which provides a variety of GUI and MCP tools. The middle part depicts the process by which the web agent handles a user query within this environment: it first retrieves relevant information from the web, then uses the tools to download and view the related paper, and finally derives the result. The right paper presents a comprehensive evaluation of the trajectory. Since part of the supporting information originates from the model's internal memory, the soundness is considered relatively weak.

$h_{t+1} = (h_t, a_t, r_t)$. The next state is then given by $s_{t+1} = T(s_t, a_t)$.

2.2 Action Space

Complex web-based tasks often require not only extensive information retrieval, but also further processing and analysis of the retrieved information. Due to the inherent hallucination problem of LLMs and their relatively weak mathematical abilities, leveraging a suite of auxiliary tools such as calculators or code executors has become a common solution.

However, performing calculations purely via UI interactions is cumbersome and may require numerous actions for a single expression; on the contrary, invoking an API-based calculator can be completed in a single step. Therefore, we design an action space that differs from previous work. Specifically, we implement a unified interaction interface supporting both GUI and MCP actions, including Playwright-based browser tools, a calculator, a file reader, and additional servers, resulting in a total of 19 tools. A complete list is provided in the Appendix A.1.

The model can issue GUI interaction commands to the browser server as well as invoke various utility tools. Our tool design also incorporates greater flexibility, enabling more diverse trajectories. Combined with a broader action space, this results in more complex trajectories that are more difficult to evaluate. At the same time, the introduction of MCP ensures strong tool extensibility, aligning with future technological developments.

2.3 Observation Space

We adopt the commonly used Accessibility Tree representation to encode the content of a web page. The Accessibility Tree preserves much of the page's structural information while being more compact than the DOM or HTML representations, thus improving the performance of the model.

3 Benchmark Construction

3.1 Task Selection

We select tasks from the widely used agent benchmark GAIA, focusing on those that involve search problems designed to test planning capabilities. GAIA is a high-quality benchmark created by human experts, designed to evaluate the comprehensive capabilities of agents, and organized into three difficulty levels.

Unlike other common web agent benchmarks that emphasize web page understanding to accomplish tasks, GAIA's samples typically require multiple supporting pieces of evidence or domain-specific knowledge, involving complex search procedures to retrieve information, comprehension of intricate real-world web content, followed by additional processing or reasoning on the retrieved information. From GAIA's validation set, we extract 95 tasks that satisfy our selection criteria.

3.2 Trajectory Generation

After collecting the tasks, we use several of the most powerful models available, including

Level	Number	Avg Trajectory Length	Round Limit	Chosen Trajectory Acc	Rejected Trajectory Acc
1	26	12.46	20	76.92%	26.92%
2	53	22.68	40	73.58%	30.19%
3	16	37.19	60	50.00%	18.75%
All	95	22.33	—	70.53%	27.37%

Table 1: Statistical information of CREAT including the number of selected tasks at each difficulty level, the average trajectory length, the round limit during execution, the accuracy of the chosen trajectories, and the accuracy of the rejected trajectories.

GPT-5 (OpenAI, 2025), Gemini-2.5-Pro (Comanici et al., 2025), DeepSeek-v3.1(DeepSeek-AI, 2024), GPT-5-mini, among others, to generate trajectories for these tasks. Each model interacts with the environment until the agent commits an answer, commits failure, or exceeds the round limit. The prompt template used for generation is provided in the Appendix A.2.

For each question, we retain exactly two trajectory: one chosen trajectory and one rejected trajectory. These are intended to reflect representative good or bad behavior exhibited by the models. Furthermore, the chosen and rejected trajectories must exhibit a noticeably quality gap, which we primarily determine based on task progress; for trajectories with similar task progress, we consider additional factors in our evaluation. If a suitable trajectory pair could not be obtained, we will continue sampling until models generate acceptable results.

We have designed a comprehensive set of evaluation dimensions for the collected trajectories, followed by manual annotation. While most previous work primarily focused on assessing the correctness of a trajectory, in practice, RL training may yield multiple trajectories that are all technically correct, and this possibility will likely increase as the agent's performance improves. Consequently, evaluating additional dimensions of trajectories to identify the optimal one is becoming increasingly important. For incorrect or failed trajectories, a comprehensive evaluation is important to diagnose the capability shortcomings of LLMs as well.

To this end, we define 5 key dimensions for holistic trajectory assessment. The design of these dimensions is informed by our observations of common errors made by agents, which reveal some current performance shortcomings of web agents. These include getting stuck in erroneous loops without adjusting strategies, hallucinating interpretations of existing information, generating redundant and meaningless steps, and reasoning based on insufficient evidence. These dimensions emphasize the overall characteristics of a trajectory rather than evaluating individual actions in isolation; for single actions, we particularly consider their impact on the trajectory as a whole or interpret their role based on contextual understanding. At the same time, we pay attention to the model's ability to adjust strategies and correct prior mistakes. Rather than using binary judgments, we define clear scoring rubrics ranging from 1 to 4 for each dimension, as detailed in the Appendix A.2.

A brief definition of each dimension is as follows: **(1) Completeness:** Measures how far the agent progresses toward the goal before the first uncorrected directional error occurs. It reflects the degree to which the trajectory reaches meaningful milestones without deviating irreversibly from the intended solution path. **(2) Adaptability:** Assesses the agent's ability to avoid and recover from stagnation or repetitive mistake patterns. This includes detecting when current strategies fail and altering its approach to avoid unproductive loops. **(3) Truthfulness:** Evaluates the presence of hallucinations, factual inaccuracies, or flawed reasoning in the trajectory, as well as the agent's capacity to amend such errors. It examines whether outputs remain factually correct and logically consistent over time. **(4) Efficiency:** Measures the proportion of redundant, unnecessary, or incorrect actions within the trajectory. It considers how much effort is wasted versus contributing to valid steps toward the goal. **(5) Soundness:** Assesses whether the agent's reasoning relies on sufficient and relevant evidence, rather than unsupported internal assumptions. It focuses on logical validity and evidence-backed decisions throughout the trajectory.

3.3 Statistical Information and Quality Verification

Following the above process, we construct CREAT, comprising 95 questions, 190 high-quality trajectories, and comprehensive manual annotations. Sta-

Model	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5	64.74%	**67.89%**	**55.26%**	39.47%	62.63%	**58.00%**	11.05%	**77.89%**
GPT-5-mini	**66.32%**	60.53%	42.63%	49.47%	69.47%	57.68%	16.32%	74.74%
GPT-5-nano	61.05%	55.79%	43.16%	44.21%	64.21%	53.68%	15.79%	72.63%
Gemini-2.5-pro	57.89%	55.79%	47.89%	42.11%	**72.11%**	55.16%	14.74%	68.42%
Gemini-2.5-flash	61.05%	55.26%	45.79%	46.84%	62.63%	54.32%	10.53%	65.26%
Gemini-2.5-flash-lite	55.79%	54.21%	38.95%	**51.05%**	66.32%	53.26%	**19.47%**	35.79%
Grok-4	**66.32%**	55.26%	52.11%	41.05%	60.00%	54.95%	13.16%	75.79%
Deepseek-v3.1-Terminus	63.16%	66.84%	43.16%	45.79%	63.68%	56.53%	17.89%	68.42%
Claude-3.7-sonnet	48.95%	47.37%	39.47%	37.89%	52.11%	45.16%	13.68%	46.32%
Claude-3.5-haiku	53.68%	48.95%	41.05%	43.68%	57.89%	49.05%	18.95%	29.47%

Table 2: Main evaluation results. Comp denotes Completeness, Adapt denotes Adaptability, Truth denotes Truthfulness, Eff denotes Efficiency, Sound denotes Soundness, All denotes the average across the 5 dimensions, Exact denotes Exact Match (i.e., agreement on all dimension scores), and Rank denotes Preference Rank, the accuracy of ranking trajectories by score. Blue cells indicate the best value for that dimension. All models perform weakly on exact match, and differences across dimensions are pronounced.

tistical information can be found in Table 1.

At every stage requiring manual annotation, including task selection, trajectory filtering, and trajectory annotation, 3 PhD-level annotators would participate, performing cross-validation to ensure the quality of the dataset.

4 Experiments

4.1 Settings

We select a total of 10 mainstream commercial models, comprising GPT-5, GPT-5-mini, GPT-5-nano, Gemini-2.5-pro, Gemini-2.5-flash, Gemini-2.5-flash-lite, Claude-3.7-sonnet (Anthropic, 2024), DeepSeek-v3.1-Terminus, Claude-3.5-haiku, and Grok-4 (xAI, 2025). Each model is provided with the evaluation criteria, the question, and the complete execution trajectory, including the accessibility tree, the URL, the agent output, and the tool response for each round, allowing the model to generate scores for each evaluation dimension.

Since these trajectories might be lengthy, which could exceed 300k tokens, we adopt a truncation strategy whereby the earliest accessibility tree is progressively omitted until the sequence length is reduced to within 128k tokens. For certain trajectories, excessive tool response lengths still cause the overall input to exceed the specified limit; in such cases, we further truncate the overly long tool responses in a gradual manner.

4.2 Results

We evaluate the 10 selected models on CREAT; the main results are shown in Table 2. We report accuracies for each of the 5 dimensions, for all dimensions combined, exact match, and preference ranking. For Exact Match, the outputs must have scores identical to human evaluations across all dimensions. For preference ranking, we rank trajectories based on the 5-dimensional scores: we first compare completeness; if completeness is tied, we compare the total score. If the chosen trajectory ranks higher than the rejected trajectory, the result is counted as correct.

Across individual dimensions, performance for all models is generally acceptable. However, exact match is low for all models , which do not exceed 20%, indicating that current LLMs may still struggle to provide accurate and comprehensive evaluations of complex trajectories.

For every single model, performance varies substantially by dimension while accuracies on truthfulness and efficiency tend to be lower across models. The top-performing model is not consistent across dimensions, suggesting pronounced differences in capabilities. Within the same model family, larger models generally perform better, but the gaps are modest; on certain dimensions, smaller variants can even perform better. To comprehensively evaluate a trajectory, it may be necessary to aggregate outputs across multiple models.

4.3 Analysis

4.3.1 Impact of Complexity

To study the impact of complexity on the accuracy of the evaluation, we report results across different levels; Table 3 shows 8 larger scale models, with additional results provided in the Appendix A.3.

For all models, the accuracy at Level 3 is sig-

Model	Level	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5	1	61.54%	76.92%	57.69%	50.00%	75.00%	64.23%	25.00%	88.46%
	2	68.87%	71.70%	58.49%	38.68%	61.32%	59.81%	7.55%	75.47%
	3	56.25%	40.62%	40.62%	25.00%	46.88%	41.88%	0.00%	68.75%
GPT-5-mini	1	61.54%	69.23%	59.62%	59.62%	80.77%	66.15%	23.08%	80.77%
	2	70.75%	61.32%	39.62%	48.11%	69.81%	57.92%	17.92%	69.81%
	3	59.38%	43.75%	25.00%	37.50%	50.00%	43.12%	0.00%	81.25%
Gemini-2.5-pro	1	57.69%	53.85%	65.38%	55.77%	82.69%	63.08%	26.92%	80.77%
	2	59.43%	60.38%	44.34%	38.68%	71.70%	54.91%	11.32%	67.92%
	3	53.12%	43.75%	31.25%	31.25%	56.25%	43.12%	6.25%	50.00%
Gemini-2.5-flash	1	63.46%	61.54%	55.77%	53.85%	75.00%	61.92%	17.31%	80.77%
	2	62.26%	57.55%	44.34%	44.34%	58.49%	53.40%	10.38%	62.26%
	3	53.12%	37.50%	34.38%	43.75%	56.25%	45.00%	0.00%	50.00%
Grok-4	1	67.31%	53.85%	63.46%	50.00%	71.15%	61.15%	21.15%	84.62%
	2	71.70%	57.55%	52.83%	43.40%	59.43%	56.98%	12.26%	75.47%
	3	46.88%	50.00%	31.25%	18.75%	43.75%	38.12%	3.12%	62.50%
DeepSeek-V3.1-Terminus	1	65.38%	71.15%	51.92%	50.00%	76.92%	63.08%	28.85%	73.08%
	2	66.04%	73.58%	45.28%	43.40%	62.26%	58.11%	16.98%	66.04%
	3	50.00%	37.50%	21.88%	46.88%	46.88%	40.62%	3.12%	68.75%
Claude-3.7-sonnet	1	57.69%	53.85%	57.69%	51.92%	44.23%	67.31%	55.00%	23.08%
	2	43.40%	50.00%	47.17%	34.91%	39.62%	49.06%	44.15%	13.21%
	3	37.50%	37.50%	31.25%	34.38%	21.88%	37.50%	32.50%	0.00%
Claude-3.5-haiku	1	61.54%	63.46%	51.92%	53.85%	61.54%	58.46%	28.85%	34.62%
	2	54.72%	46.23%	38.68%	39.62%	61.32%	48.11%	18.87%	26.42%
	3	37.50%	34.38%	31.25%	40.62%	40.62%	36.88%	3.12%	31.25%

Table 3: Results of complexity analysis. Comp denotes Completeness, Adapt denotes Adaptability, Truth denotes Truthfulness, Eff denotes Efficiency, Sound denotes Soundness, All denotes the average score across the 5 dimensions, Exact refers to Exact Match, and Rank refers to Preference Ranking, i.e., the accuracy of ranking trajectories according to their scores. The red cells indicate a difference of more than 10% compared with Level 1. All models exhibit clear performance declines as the level increases, with notable differences among dimensions.

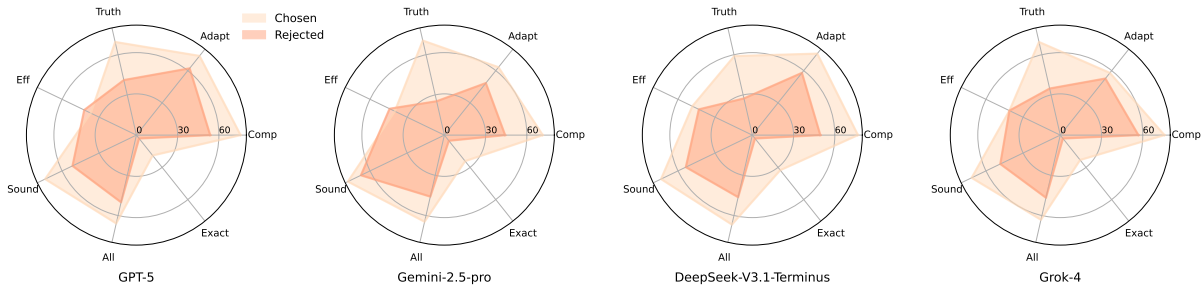
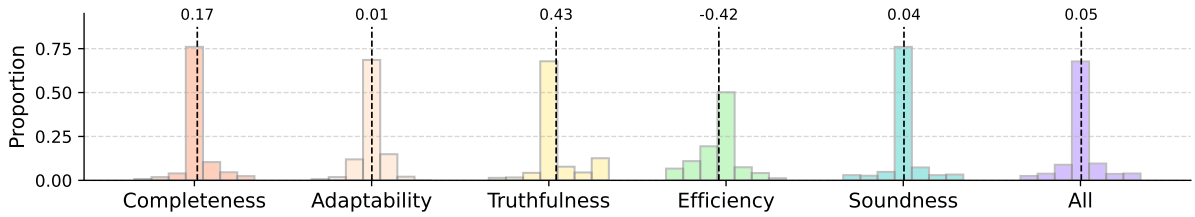


Figure 3: Results by trajectory type. Metrics include completeness (Comp), adaptability (Adapt), truthfulness (Truth), efficiency (Eff), soundness (Sound), as well as their average (All) and exact match agreement across all dimensions (Exact). Apart from efficiency, declines are pronounced across other dimensions.

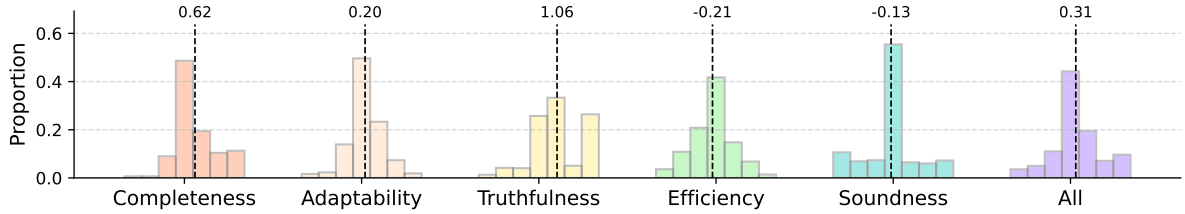
nificantly lower than at Level 1 in almost all aspects, with differences exceeding 20% in a substantial number of cases. Moreover, the average score across the 5 dimensions steadily decreases as the level increases, indicating limitations in the LLM's ability to handle lengthy and complex trajectories as a judge.

For Preference Ranking, accuracy likewise tends to drop significantly as the level increases. This implies that the declines in accuracy in other dimensions are not solely due to a global shift in scoring, such as assigning generally lower scores, but rather reflect a genuine degradation in judgment ability when faced with more complex trajectories.

It is noteworthy that, between Gemini, which has a 1M context length, and GPT-5, which has an



(a) Distribution of chosen trajectories



(b) Distribution of rejected trajectories

Figure 4: Distribution of differences between model and human scores. The upper plot shows chosen trajectories; the lower plot shows rejected trajectories. Each dimension shares the same horizontal axis. For each dimension, values run from -3 to 3 from left to right; the vertical axis denotes the proportion at each value. The black dashed line indicates the mean of differences.

input limit of 288k tokens, Gemini exhibits lower 5 dimension averages at Level 1 and Level 2, but performs better at Level 3, even though the maximum input length of 128k tokens is well below the context limits of both models. This suggests that enhancing a model's long text comprehension capability can, to some extent, improve its performance in this scenario.

4.3.2 Impact of Trajectory Quality

We compute accuracies separately for chosen trajectories and rejected trajectories; partial results are shown in Figure 3, with additional results in the Appendix A.3. For all dimensions except efficiency, accuracy on rejected trajectories declines substantially compared to chosen trajectories. Efficiency is comparable and in some cases even better. Exact match drops to nearly 0, indicating that models are much less accurate and more inconsistent when evaluating lower-quality trajectories, which drives the poor exact match performance.

4.3.3 Bias Analysis

To further analyze the causes of performance differences across trajectory quality, we plot the distribution of score differences between the 5 models and human evaluators (negative values indicate underestimation, positive values indicate overestimation), as shown in Figure 4. The results reveal that for chosen trajectories, the models tend to overestimate Truthfulness. Conversely, for rejected trajectories, there is a more pronounced overestimation of Truthfulness and Completeness. Models also tend to generally underestimate the efficiency of trajectories.

Strategy	All	Exact
default	45.64%	2.56%
truncate earliest rounds	47.69%	5.13%
w/o truncation	49.74%	5.13%

Table 4: Test results of Gemini-2.5-pro under different truncation strategies. All denotes the average score across the 5 dimensions, and Exact refers to Exact Match. Accuracy decreases slightly under truncation, with the "truncate earliest rounds" strategy performing marginally better.

4.3.4 Truncation Strategy

We select the model with the longest context window, Gemini-2.5-pro, and trajectories exceeding 128k tokens to compare different truncation strategies, including our default strategy, truncating the earliest rounds, and using the full trajectory. The results are shown in Table 4. It can be observed that truncation leads to a slight decrease in accuracy, with the "truncate earliest rounds" strategy incurring comparatively smaller performance loss. This suggests that retaining the complete informa-

tion from the later rounds is more important for trajectory evaluation, and that the model's capability may still be insufficient to fully utilize the additional information contained in a complete trajectory.

4.3.5 Key Findings and Error Analysis

Through a qualitative examination of the LLM's failure cases, we identify two recurring categories of errors, which align closely with our quantitative observations in the experiments:

Hallucination unawareness. When intermediate results contain hallucinations or when the reasoning trajectory is flawed, the LLM often fails to detect the errors. This issue is particularly pronounced in cases where hallucinated intermediate results are produced with high confidence and never corrected, ultimately leading to incorrect final answers. In such cases, the model tends to misinterpret these faulty steps as valid progress, resulting in systematic overestimation of completeness and truthfulness.

Misunderstanding exploration. The model struggles to distinguish between exploratory actions (necessary attempts in information-sparse scenarios) and redundant actions (truly wasteful and non-informative moves), especially in complex tasks that involve substantial search space. Since exploratory actions may superficially resemble redundant ones in terms of observable task progress, the model frequently underestimates efficiency across such cases.

5 Related Work

Web Agents. Controlling agents to perform autonomous web navigation via natural language has long been a research area of considerable interest. With the rapid development of large language models, web agents based on LLMs and MLLMs have become mainstream. Nakano et al. constructed a text-based web environment and fine-tuned GPT-3 to enable web search and question answering based on retrieved information. Similarly, Chowdhery et al. employed Flan-U-PaLM to generate Python code for interacting with the environment. Other studies have explored approaches using screenshot inputs (Lee et al., 2023), as well as preprocessing screenshots by adding masks and labels to interactive elements (He et al., 2024). More recent work has increasingly favored RL as a means to further enhance model performance (Qi et al., 2025; Wei et al., 2025).

Reward Benchmarks and Web Agent Evaluation. There has been substantial work on evaluating reward models across various scenarios, both in text-only settings such as chat and reasoning (Lambert et al., 2024; Malik et al., 2025; He et al., 2025), and in multi-modal contexts (Li et al., 2025). Research into reward benchmarks for agents has also emerged, encompassing both action-wise (Men et al., 2025) and trajectory-wise (Lù et al., 2025) evaluations.

Fo the evaluation of web agent, earlier work primarily evaluated these agents on simplified web environments (Liu et al., 2018). To study their performance in more authentic settings, researchers have constructed benchmarks based on real offline websites (Deng et al., 2023), simulated real websites (Zhou et al., 2024), and real web environments (Pan et al., 2024b). More recently, there has been a surging interest in building more challenging benchmarks for information retrieval tasks (Wu et al., 2025; Wei et al., 2025; Miyai et al., 2025). For such tasks, the resulting trajectories are often lengthy and complex. However, existing studies lack comprehensive evaluations for complex trajectories, and action-level assessments frequently overlook the implicit cumulative effects of individual actions on the overall trajectory. Our proposed CREAT is designed to address this gap.

6 Conclusion

In this work, we introduce CREAT, a benchmark specifically designed to assess the capabilities of LLM-based judges across multiple evaluation dimensions. CREAT consists of 95 questions and 190 trajectories generated by mainstream models, annotated within a 5-dimensional framework covering completeness, adaptability, truthfulness, efficiency, and soundness. Through experiments on 10 representative LLMs, we reveal the limitations of LLMs in judging complex trajectories, as well as their inherent biases. Our in-depth analysis further demonstrates their insufficient ability to detect complex hallucinations and to differentiate exploratory actions from redundant ones. We believe that our findings provide a valuable foundation for advancing both the reliability of LLM-based trajectory evaluation and the development of higher-performance web agents in real-world environments.

Limitations

Despite the contributions of this work, several limitations should be acknowledged. First, due to time constraints, we are unable to conduct experiments on open-source LLMs such as Qwen and Llama. As a result, the current evaluation results only reflect the performance of mainstream commercial models. Second, the number of available tools for trajectory generation in our current implementation remains limited, with only 19 tools integrated into the system, potentially affecting the generalizability of our findings to real-world scenarios with richer toolsets. In the future, we will consider testing more models and incorporating a larger set of tools to generate more complex trajectories.

References

Anthropic. 2024. Introducing the next generation of claude.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Preprint*, arXiv:2406.11896.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Preprint*, arXiv:2306.06070.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *Preprint*, arXiv:2401.13919.

Qi He, Cheng Qian, Xiusi Chen, Bingxiang He, Yi R., Fung, and Heng Ji. 2025. Veri-r1: Toward precise and faithful claim verification via online reinforcement learning. *Preprint*, arXiv:2510.01932.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. *Preprint*, arXiv:2210.03347.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. 2025. Vl-rewardbench: A challenging benchmark for vision-language generative reward models. *Preprint*, arXiv:2411.17451.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. *Preprint*, arXiv:1802.08802.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. 2025. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories. *Preprint*, arXiv:2504.08942.

Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. 2024. Laser: Llm agent with state-space exploration for web navigation. *Preprint*, arXiv:2309.08172.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. *Preprint*, arXiv:2506.01937.

Tianyi Men, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Agent-rewardbench: Towards a unified benchmark for reward modeling across perception, planning, and safety in real-world multimodal agents. *Preprint*, arXiv:2506.21252.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Atsuyuki Miyai, Zaiying Zhao, Kazuki Egashira, Atsuki Sato, Tatsumi Sunada, Shota Onohara, Hiromasa Yamanishi, Mashiro Toyooka, Kunato Nishina, Ryoma Maeda, Kiyoharu Aizawa, and Toshihiko Yamasaki. 2025. Webchorearena: Evaluating web browsing agents on realistic tedious web tasks. *Preprint*, arXiv:2506.01952.

Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. 2025. Nnetnav: Unsupervised learning of browser agents through environment interaction in the wild. *Preprint*, arXiv:2410.02907.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. Introducing gpt-5.

Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024a. Autonomous evaluation and refinement of digital agents. *Preprint*, arXiv:2404.06474.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. 2024b. Webcanvas: Benchmarking web agents in online environments. *Preprint*, arXiv:2406.12373.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. 2025. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *Preprint*, arXiv:2411.02337.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. *Preprint*, arXiv:2403.02502.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, Wanjun Zhong, Yining Ye, Yujia Qin, Yuwen Xiong, Yuxin Song, Zhiyong Wu, Aoyan Li, Bo Li, Chen Dun, and 93 others. 2025. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *Preprint*, arXiv:2509.02544.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. 2025. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *Preprint*, arXiv:2505.16421.

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

xAI. 2025. Grok 4 model card.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025. Aguvis: Unified pure vision agents for autonomous gui interaction. *Preprint*, arXiv:2412.04454.

Yuanzhao Zhai, Tingkai Yang, Kele Xu, Feng Dawei, Cheng Yang, Bo Ding, and Huaimin Wang. 2024. Enhancing decision-making for llm agents via step-level q-value models. *Preprint*, arXiv:2409.09345.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

Server	Tool	Type	Description
Browser	click(element_id)	GUI	Click at an element by ID
	hover(element_id)	GUI	Hover on an element by ID
	type(text, element_id, enter_flag)	GUI	Type to an element by ID
	goto(url)	MCP	Navigate the browser to a specific URL
	go_back	GUI	Navigate to the previous page
	go_forward	GUI	Navigate to the next page, if a previous 'go_back' was performed
	scroll(direction)	GUI	Scroll the page vertically
	new_tab	GUI	Create a new tab
	tab_focus(page_number)	GUI	Focus on a specific tab
	close_tab	GUI	Close the current tab
Task Manager	commit_failed	MCP	Commit task failed
	commit_completed	MCP	Commit task completed
	commit_answer(answer)	MCP	Commit answer if the task requires
Calculator	calc_expression(expression)	MCP	Evaluates a Python-style mathematical expression
	get_statistic(array, stat_type)	MCP	Calculate a specific statistic from an array on the stat_type parameter
File Reader	convert_to_markdown(file_path)	MCP	Reads a file and converts it into markdown format
Downloader	download(url)	MCP	Download a file and save to the local filesystem
Note	add_note(content)	MCP	Add the content to the note, which will show in the history
Python Executor	run_python(code)	MCP	Execute python code

Table 5: All GUI and MCP tool servers included in the action space with their corresponding supported functions, in both GUI operation and MCP, amounting to a total of 7 servers and 19 tools, each accompanied by a concise description of its functionality.

A Appendix

A.1 Action Space

This section presents detailed information regarding the design of the action space. Specifically, we implement a total of 7 MCP servers and 19 tools, supporting both GUI operations and API interactions. Relevant information is provided in Table 5. The Browser includes all 9 GUI tools as well as 1 MCP tool. Our design follows the conventions of most existing web-agent benchmarks, enabling the agent to simulate nearly all human operations that occur in realistic scenarios.

In addition, we implement a Task Manager that allows the model to autonomously manage task progress. The Downloader completes download tasks, including obtaining images, papers, and other materials. Combined with the File Reader, various file types are converted into plain-text Markdown format. For images, captions can be generated by file reader to assist the agent in multimodal understanding tasks. The Note tool can be used to record key information, while the Calculator and Python Executor are designed for further processing and analysis of data. Through this streamlined design, the model is capable of tackling a wide range of complex problems involving multimodal information.

A.2 Prompt

This section contains the prompts used for benchmark construction and evaluation, as shown in Tables 6 and 7. Our prompt designs are crafted by human experts, while also drawing upon prior work.

Table 6 provides the system prompt used to generate trajectories. We begin by introducing the general task setting, followed by specifying the types of information provided to the model at each step. Next, we describe the available MCP servers, and then present a detailed task description. To ensure the high quality of trajectories, we also compile common typical errors into hints to guide the model. Finally, we define the required output format.

For evaluation, as shown in Table 7, the tasks are similarly introduced, accompanied by comprehensive scoring criteria, followed by the prescribed output format. Since models may not strictly adhere to the format, we employ regular expression matching to extract results; any sample with incomplete matches is considered incorrect.

In summary, while concise, our prompt design fully and clearly describes the tasks, and has been tested to ensure its effectiveness.

A.3 Additional Results

We test a total of 10 models; however, the analysis section does not present the complete results.

Prompt for trajectory generation

You are a web agent responsible for completing the user's objective using tools provided as MCP servers. One of them is the browser MCP server for web interactions.
At each step, you will be given:
1. The user's objective 2. The current webpage's accessibility tree 3. The current webpage's URL 4. The full history of prior MCP interactions 5. A step by step plan
Available Tools
Tool list starts with '## Available Tools'. Each server starts with '### server_name', followed by its tools, each starting with '- tool_name'. The description of each tool includes: - Function summary - Input parameters (name, type, required status, description)
...
Your task
Your goal is to carefully reason about the current progress toward the user's objective based on all provided information (objective, webpage state, URL, history, plan). Then, output the next action to advance toward the objective, selecting the most suitable MCP server and tool based on context.
1. Analyze the current situation and determine how far the task has progressed and what has been accomplished so far.
2. Reason about the specific next action that should be taken to advance toward completion.
3. Output **exactly one MCP tool call** that represents this next action, selecting the most appropriate tool and parameters based on context.
Important Notes (Pay special attention to when handling tasks.)
1. The Accessibility Tree only contains the currently visible web page content. To browse the entire page, you need to scroll the page.
2. The Accessibility Tree also contains the opened tabs.
3. For **starting a new search**, first navigate to the homepage (e.g., goto(www.bing.com)) to avoid multiple search queries in one search bar.
4. If you believe the task has failed or is complete, use the 'task manager' to commit.
5. For **important content**, such as **intermediate results**, please use the 'note' tool to record.
6. For PDF file, download to local file system, and convert to markdown to view.
Output Format
Thought: ...
Action: <tool>
{
"server_name": "",
"tool_name": "",
"inputs": {
"<parameter1>": "<value1>",
"<parameter2>": "<value2>",
...
}
}
</tool>

Table 6: Prompt for trajectory generation.

Here, we provide the remaining results in Tables 3 and 9, including the performance of GPT-nano and Gemini-2.5-flash-lite across trajectories of different complexity levels, as well as the performance of six models on trajectories of varying quality. Overall, the observed performance patterns are consistent with the findings reported in the main text. Due to space limitations, these results are excluded from the main body.

Prompt for trajectory evaluation

You are tasked with evaluating the performance of an LLM-agent based on a given trajectory for a specific task. The trajectory consists of multiple turns, each containing the following components: Accessibility Tree, Current URL, Agent Output, and Tool Response. The agent's goal is to complete the task using available MCP tools, including a browser.
Your evaluation should be based on the following criteria:
1. **Completeness**: Assess how far the agent progresses towards the task goal before making an uncorrected, direction-changing error.
- 4 points: Task completed correctly.
- 3 points: Task > 2/3 complete but not finished.
- 2 points: Task between 1/3 and 2/3 complete.
- 1 point: Task < 1/3 complete.
2. **Adaptability**: Evaluate the agent's ability to avoid being stuck in repetitive errors or stalled progress, and whether it can adjust its strategy effectively.
- 4 points: Never gets stuck in an error loop or progress stall.
- 3 points: Gets stuck but successfully adjusts strategy, leading to significant correct progress.
- 2 points: Gets stuck, adjusts strategy, but progress is minimal or incorrect.
- 1 point: Gets stuck without adjusting strategy; repeats ineffective attempts.
3. **Truthfulness**: Check for hallucinations or factual/reasoning errors in the trajectory.
- 4 points: No hallucinations or errors.
- 3 points: Minor hallucinations/errors with negligible impact, or quickly corrected.
- 2 points: Significant errors impacting execution, requiring many steps to correct.
- 1 point: Serious errors impacting execution, remaining uncorrected, or leading to a wrong final answer.
4. **Efficiency**: Measure the proportion of redundant or hallucinated actions.
- 4 points: No redundant steps.
- 3 points: Redundant steps $\leq$ 10% of total or $\leq$ 1 step.
- 2 points: Redundant steps $\leq$ 20% of total or $\leq$ 3 steps.
- 1 point: Redundant steps > 20% of total.
5. **Soundness**: Assess whether the agent verifies critical, non-common-sense information before using it to produce the answer.
- 4 points: All critical information is verified; if progress stalls but no unverified usage occurs, score is 4.
- 3 points: Half or fewer of the critical information pieces are unverified.
- 2 points: More than half of the critical information is unverified.
- 1 point: Ignores critical information gathering or hallucinates such information.
Output Format:
Provide your scores in the following format:
Completeness: <score>
Adaptability: <score>
Truthfulness: <score>
Efficiency: <score>
Soundness: <score>

Table 7: Prompt for trajectory evaluation.

Model	Level	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5-nano	1	63.46%	61.54%	57.69%	53.85%	80.77%	63.46%	26.92%	73.08%
	2	64.15%	61.32%	42.45%	43.40%	61.32%	54.53%	14.15%	69.81%
	3	46.88%	28.12%	21.88%	31.25%	46.88%	35.00%	3.12%	81.25%
Gemini-2.5-flash-lite	1	57.69%	59.62%	46.15%	51.92%	75.00%	58.08%	28.85%	26.92%
	2	57.55%	52.83%	40.57%	48.11%	68.87%	53.58%	19.81%	39.62%
	3	46.88%	50.00%	21.88%	59.38%	43.75%	44.38%	3.12%	37.50%

Table 8: Additional Results of complexity analysis. Comp denotes Completeness, Adapt denotes Adaptability, Truth denotes Truthfulness, Eff denotes Efficiency, Sound denotes Soundness, All denotes the average score across the 5 dimensions, Exact refers to Exact Match, i.e., all dimension scores matching human ratings, and Rank refers to Preference Ranking, i.e., the accuracy of ranking trajectories according to their scores. All models exhibit clear performance declines as the level increases, with notable differences among dimensions.

Model	type	Comp	Adapt	Truth	Eff	Sound	All	Exact
Claude-3.7-sonnet	Chosen	62.11%	57.89%	54.74%	42.11%	63.16%	56.00%	27.37%
	Reject	35.79%	36.84%	24.21%	33.68%	41.05%	34.32%	0.00%
Claude-3.5-haiku	chosen	69.47%	60.00%	65.26%	58.95%	69.47%	64.63%	37.89%
	rejected	37.89%	37.89%	16.84%	28.42%	46.32%	33.47%	0.00%
GPT-5-mini	chosen	82.11%	71.58%	66.32%	51.58%	80.00%	70.32%	31.58%
	rejected	50.53%	49.47%	18.95%	47.37%	58.95%	45.05%	1.05%
GPT-5-nano	chosen	72.63%	71.58%	65.26%	52.63%	74.74%	67.37%	30.53%
	rejected	49.47%	40.00%	21.05%	35.79%	53.68%	40.00%	1.05%
Gemini-2.5-flash	chosen	74.74%	61.05%	67.37%	45.26%	71.58%	64.00%	20.00%
	rejected	47.37%	49.47%	24.21%	48.42%	53.68%	44.63%	1.05%
Gemini-2.5-flash-lite	chosen	69.47%	66.32%	64.21%	65.26%	74.74%	68.00%	38.95%
	rejected	42.11%	42.11%	13.68%	36.84%	57.89%	38.53%	0.00%

Table 9: Additional results for different trajectory types. Comp denotes Completeness, Adapt denotes Adaptability, Truth denotes Truthfulness, Eff denotes Efficiency, Sound denotes Soundness, All denotes the average score across the 5 dimensions, and Exact refers to Exact Match, i.e., all dimension scores matching human ratings. Apart from Efficiency, declines in other dimensions are notably apparent.