

CREAT: A Comprehensive Reward Benchmark 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 while 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**—a Comprehensive **RE**ward benchmark for lengthy and complex web **Agent** Trajectories. CREAT is not only a benchmark containing challenging, high-order web browsing queries that demand long-horizon agentic reasoning, but also a comprehensive, fine-grained evaluation framework for assessing agent trajectories. It evaluates trajectories along five dimensions that are crucial for web agents, going beyond correctness alone. Experiments on 10 representative 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) (DeepSeek-AI et al., 2025; OpenAI, 2025; Huang et al., 2025a) and Multimodal Large Language Models (MLLMs) (Liu et al., 2023; OpenAI et al., 2024; Bai et al., 2025; Su et al., 2025) have made it increasingly feasible to develop powerful web agents capable of interacting with web browser (Nakano et al., 2022;

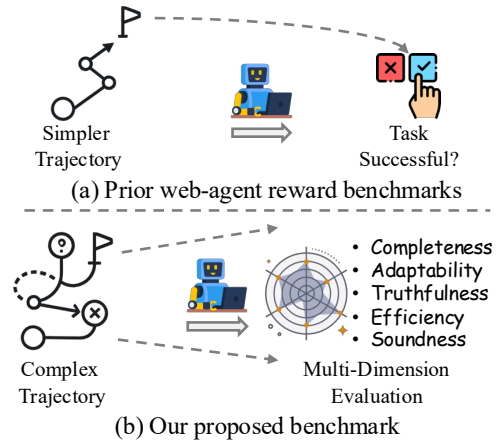


Figure 1: Comparison between (a) prior web-agent reward benchmarks, which largely emphasize simple tasks and correctness-only scoring, and (b) our CREAT benchmark, which targets long-horizon, complex trajectories and multi-dimensional evaluation.

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).

To further enhance the performance of web agent, we introduce **CREAT** — a Comprehensive **RE**ward benchmark for lengthy and complex web **Agent** Trajectories — designed to evaluate the capabilities of LLM-based judges in realistic, challenging scenarios. 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; Huang et al., 2025b). Recent work shows that LLMs can serve as automatic

¹Our data and code will be publicly released.

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. Existing reward benchmarks (Lù et al., 2025) largely focus on relatively simple tasks that require minimal strategy, or they primarily assess trajectory correctness (success/failure) as shown 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 motivates our core question: **Can LLMs serve as reliable judges for comprehensively evaluating lengthy, complex web-agent trajectories?**

To address this, CREAT selects 100 high-complexity tasks 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, calculator). Using this environment, we employ strong commercial models to collect high-quality trajectories and evaluate them across **five dimensions**: (1) **Completeness**: How much of the task has been correctly completed. (2) **Adaptability**: The agent’s ability to avoid stagnation and adjust when progress stalls. (3) **Truthfulness**: The extent of hallucinations in the trajectory and the agent’s ability to correct them. (4) **Efficiency**: The proportion of redundant steps in the trajectory. (5) **Soundness**: The agent’s ability to gather key information to support reasoning and reduce reliance on internal knowledge. The annotation process is conducted by human annotators, following a rigorous quality control. From a large pool of candidate trajectories, we select and retain 200 representative examples to 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 100 complex problems that demand high-order web-browsing operations and long-horizon agentic reasoning, together with 200 high-quality trajectories. It further introduces five fine-grained evaluation dimensions aligned with practical error types, with well-defined scoring criteria. It also provides annotations with justifications and additional fine-grained supplementary information.
 - 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

To construct a challenging benchmark that meets future needs, we 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. The transition function $T : S \times A \rightarrow S$ is deterministic. The plan P specifies the execution strategy and the main steps for accomplishing the task. Details of the plan generation are provided in Appendix A.2.

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

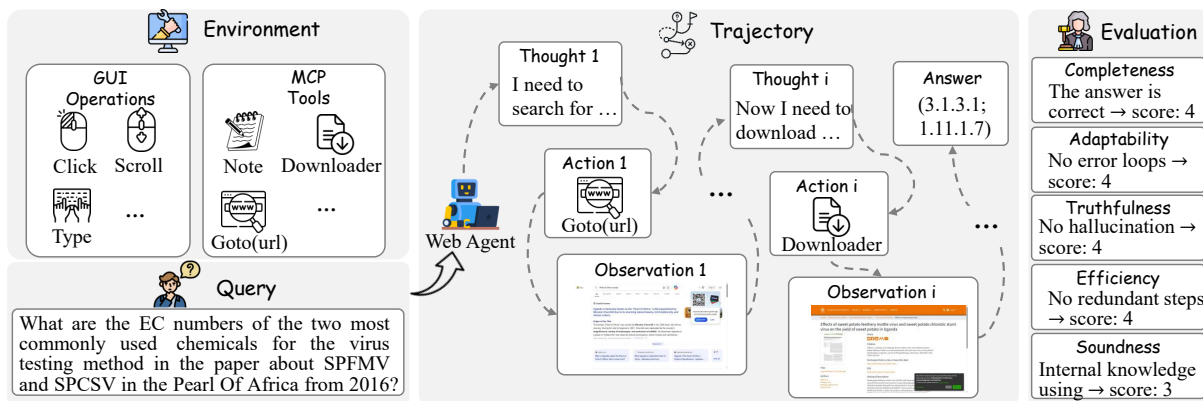


Figure 2: Overview of CREAT. The left part illustrates the unified environment we implemented, which provides a variety of GUI and MCP tools. The middle part depicts the process by which the web agent handles a user query within this environment: it first retrieves relevant information from the web, then uses the tools to download and view the related paper, and finally derives the result. The right part 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.

Accessibility Tree and its URL, while h_t denotes the history of interactions with tool servers, i.e., $h_{t+1} = (h_t, a_t, r_t)$. 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 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 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 a browser, 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 MCP 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 (Mialon et al., 2023), 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 100 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 Turns	Turn Limit	Chosen Trajectory Acc	Rejected Trajectory Acc
1	26	12.63	20	80.77%	23.08%
2	56	22.25	40	70.18%	28.07%
3	17	37.71	60	58.82%	11.76%
All	100	22.38	—	71.00%	24.00%

Table 1: Statistical information of CREAT, including the number of selected tasks at each difficulty level, the average trajectory turns, the turn limit during execution, the accuracy of the chosen trajectories, and the accuracy of the rejected trajectories. An example for each difficulty level is shown in Table 13.

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.3.

For each question, we retain exactly two trajectories: 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, reinforcement learning (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 interpreta-

tions 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.

A brief definition of each dimension is as follows: **(1) Completeness:** How much of the task has been completed, approximated as (correctly executed steps) / (estimated total steps needed). **(2) Adaptability:** Assesses the agent’s ability to avoid and recover from stagnation. 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. **(5) Soundness:** The agent’s ability to obtain key external information to support its answer and reduce reliance on internal knowledge. The agent should gather crucial information (non-common-knowledge obtained from the web) and reason over it. Table 10 provides the instruction given to the annotators, including specific definitions and detailed scoring criteria.

3.3 Statistical Information and Quality Verification

Following the above process, we construct CREAT, comprising 100 questions, 200 high-quality trajectories, and comprehensive manual annotations. Statistical information can be found in Table 1.

Model	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5	61.00	57.00	43.00	50.50	50.50	52.40	16.00	69.00
GPT-5-mini	53.00	52.50	41.00	49.50	54.50	50.10	17.00	65.00
Gemini-2.5-pro	60.50	61.50	46.50	47.50	75.00	58.20	15.00	72.00
Gemini-2.5-flash	60.50	67.00	42.00	53.00	65.00	57.50	17.00	62.00
Grok-4	61.00	52.50	48.00	45.50	64.50	54.30	15.00	76.00
Claude-sonnet-4.5	60.00	66.50	47.00	45.00	45.50	52.80	13.50	70.00
DeepSeek-V3.2	60.50	57.00	38.00	48.00	54.50	51.60	14.50	70.00
Kimi-K2	61.00	57.50	44.50	52.00	61.50	55.30	19.00	48.00
Qwen3-30b-a3b	59.50	61.00	37.50	54.50	57.50	54.00	19.50	31.00
Llama3.3-8b	55.00	56.50	40.50	45.00	68.50	53.10	17.50	34.00

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.

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. For each annotation, we provide a justification and additional fine-grained supplementary information. The detailed annotation process and an example of annotation are provided in Appendix A.4.

4 Experiments

4.1 Settings

We select a total of 10 mainstream models, comprising GPT-5 (OpenAI, 2025), GPT-5-mini, Gemini-2.5-pro (Comanici et al., 2025), Gemini-2.5-flash, Grok-4 (xAI, 2025), Claude-sonnet-4.5 (Anthropic, 2025), DeepSeek-v3.2 (DeepSeek-AI, 2024), Kimi-K2 (Team et al., 2025), Qwen3-30b-a3b (Yang et al., 2025) and LLaMA3.3-8b (Grattafiori et al., 2024). Models will be provided with the scoring criteria, the question, and the complete trajectory. The full prompt is shown in Table 9. For each input, the model will directly output a score for each dimension. Since some trajectories are extremely lengthy which could exceed 300k tokens while the majority of models we selected have a context length of 128k, we truncate the earliest turns that exceed the 128k-token limit.

4.2 Results

We evaluate the 10 selected models on CREAT. For each dimension, the accuracy is calculated by comparing the model-generated scores with the

human-annotated scores. The results are presented in Table 2. We also report accuracies 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. As the completeness is the most crucial for agent employment, we would 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, around 50% or below. The top-performing model is not consistent across dimensions, suggesting pronounced differences in capabilities. Larger models generally perform better, though the margin is not substantial. In certain dimensions, smaller models may even outperform. To comprehensively evaluate a trajectory, it may be necessary to aggregate outputs across multiple models. In the case of Preference Ranking, smaller models perform markedly worse. For example, Qwen and LLaMA achieve only around 30% indicating a diminished capability in evaluation of overall quality.

Model	Level	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5	1	67.31	61.54	55.77	61.54	63.46	61.92	28.85	73.08
	2	59.65	58.77	40.35	48.25	46.49	50.70	14.04	64.91
	3	55.88	44.12	32.35	41.18	44.12	43.53	2.94	76.47
Gemini-2.5-pro	1	61.54	65.38	61.54	55.77	90.38	66.92	25.00	80.77
	2	61.40	63.16	42.11	49.12	72.81	57.72	14.04	71.93
	3	55.88	50.00	38.24	29.41	58.82	46.47	2.94	58.82
DeepSeek-V3.2	1	65.38	67.31	42.31	57.69	65.38	59.62	25.00	61.54
	2	62.28	54.39	35.96	47.37	52.63	50.53	12.28	75.44
	3	47.06	50.00	38.24	35.29	44.12	42.94	5.88	64.71
Claude-sonnet-4.5	1	65.38	71.15	50.00	51.92	53.85	58.46	21.15	61.54
	2	59.65	64.91	47.37	43.86	44.74	52.11	14.04	71.93
	3	52.94	64.71	41.18	38.24	35.29	46.47	0.00	76.47
Qwen3-30b-a3b	1	63.46	73.08	44.23	65.38	61.54	61.54	30.77	38.46
	2	60.53	60.53	35.09	51.75	61.40	53.86	18.42	26.32
	3	50.00	44.12	35.29	47.06	38.24	42.94	5.88	35.29
Llama3.3-8b	1	59.62	65.38	53.85	53.85	71.15	60.77	26.92	26.92
	2	57.89	57.02	37.72	44.74	72.81	54.04	17.54	38.60
	3	38.24	41.18	29.41	32.35	50.00	38.24	2.94	29.41

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.

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 6 models, with full results provided in the Table 14.

For all models, the accuracy at Level 3 is significantly lower than at Level 1 in almost all aspects, with differences exceeding 10% in a substantial number of cases. Moreover, the average score 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, as the level increases, the model’s accuracy exhibits no clear trend. Notably, Gemini, which possesses a 1M-token context length, achieves substantially higher overall accuracy than GPT-5, whose context length is 288k. However, Gemini experiences a marked drop in accuracy for Preference Ranking at Level 3, despite the largest input length of 128k tokens being far smaller than its full 1M-token capacity. In contrast, GPT-5 demonstrates significantly better accuracy

in Preference Ranking at Level 3. This implies that, for fine-grained evaluation, the model’s ability to understand long contexts may be critical, whereas for assessment of overall quality, such capacity might play a less decisive role.

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 full results in Table 15. 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 10 models and human evaluators (negative values indicate underes-

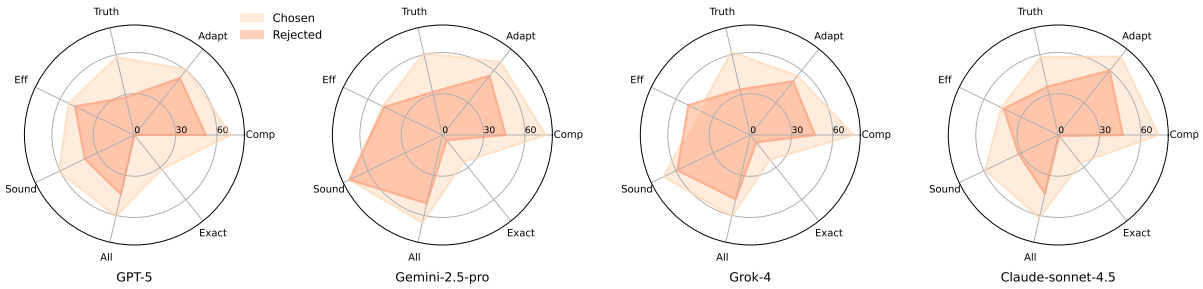
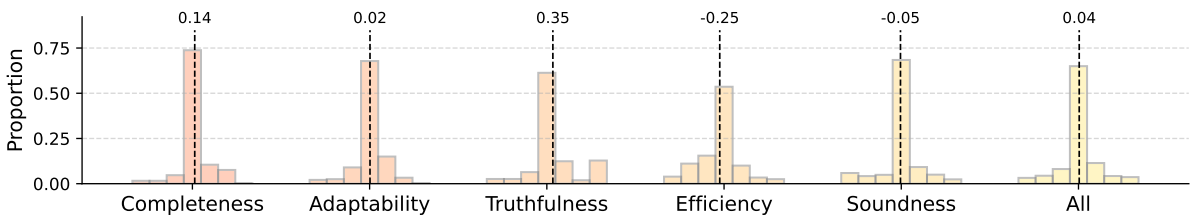
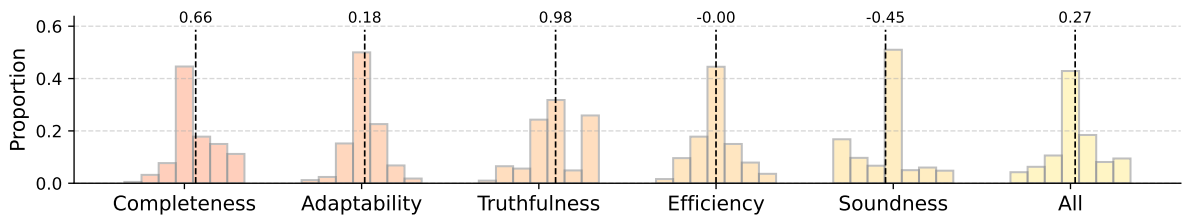


Figure 3: Results by trajectory type. Metrics include completeness (Comp), adaptability (Adapt), truthfulness (Truth), efficiency (Eff), soundness (Sound), as well as their average (All) and exact match agreement across all dimensions (Exact). Apart from efficiency, declines are pronounced across other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



(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.

418 timation, positive values indicate overestimation),
 419 as shown in Figure 4. The results reveal that for
 420 chosen trajectories, the models tend to overestimate
 421 Truthfulness. For rejected trajectories, there is a
 422 more pronounced overestimation of Truthfulness
 423 and Completeness. Models also tend to generally
 424 underestimate the efficiency of chosen trajectories.

4.3.4 Truncation Strategy

425 We select the model with the longest context win-
 426 dow, Gemini-2.5-pro, and trajectories exceeding
 427 128k tokens to compare different truncation strat-
 428 egies, including our default strategy, fine-grained
 429 truncation strategy, and using the full trajectory.
 430 The fine-grained truncation strategy refers to pro-
 431 gressively removing the earliest observations and
 432 tool responses from the trajectory while retaining
 433

Strategy	All	Exact
default	48.57%	4.76%
fine-grained truncation	50.95%	7.14%
w/o truncation	51.43%	4.76%

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 fine-grained truncation strategy performing marginally better.

434 the actions and thoughts. The results are shown in
 435 Table 4. It can be observed that truncation leads to
 436 a slight decrease in accuracy, with the fine-grained
 437 truncation strategy incurring comparatively smaller
 438 performance loss. This suggests that the impor-

tance of observations and tool responses is relatively limited, 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 with a typical example provided in Appendix A.5, 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. (2022) 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.

For 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 comprehensively evaluate the capability of LLM judges to perform fine-grained assessments of complex agent trajectories. CREAT comprises 100 questions and 200 high-quality trajectories, along with a fine-grained evaluation framework encompassing five dimensions: completeness, adaptability, truthfulness, efficiency, and soundness. Through experiments conducted on 10 representative LLMs, we reveal their limitations in assessing lengthy trajectories as well as several biases, including insufficient ability to detect complex hallucinations and to distinguish exploratory actions from redundant ones. We believe that our findings can contribute to improving the performance and reliability of LLM judges, thereby enabling web agents to better adapt to increasingly complex real-world scenarios in the future.

536 Limitations

537 Despite the contributions of this work, several limi-
538 tations should be acknowledged. First, due to time
539 constraints, we are unable to conduct experiments
540 on open-source LLMs such as Qwen and Llama.
541 As a result, the current evaluation results only re-
542 flect the performance of mainstream commercial
543 models. Second, the number of available tools for
544 trajectory generation in our current implementa-
545 tion remains limited, with only 19 tools integrated
546 into the system, potentially affecting the generaliz-
547 ability of our findings to real-world scenarios with
548 richer toolsets. In the future, we will consider test-
549 ing more models and incorporating a larger set of
550 tools to generate more complex trajectories.

551 References

552 Anthropic. 2025. [Introducing claude sonnet 4.5](#).

553 Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane
554 Suhr, Sergey Levine, and Aviral Kumar. 2024. [Di-
555 girl: Training in-the-wild device-control agents
556 with autonomous reinforcement learning](#). *Preprint*,
557 arXiv:2406.11896.

558 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wen-
559 bin Ge, Sibao Song, Kai Dang, Peng Wang, Shi-
560 jie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu,
561 Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei
562 Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 oth-
563 ers. 2025. [Qwen2.5-vl technical report](#). *Preprint*,
564 arXiv:2502.13923.

565 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin,
566 Maarten Bosma, Gaurav Mishra, Adam Roberts,
567 Paul Barham, Hyung Won Chung, Charles Sutton,
568 Sebastian Gehrmann, Parker Schuh, Kensen Shi,
569 Sasha Tsvyashchenko, Joshua Maynez, Abhishek
570 Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vin-
571 odkumar Prabhakaran, and 48 others. 2022. [Palm:
572 Scaling language modeling with pathways](#). *Preprint*,
573 arXiv:2204.02311.

574 Gheorghe Comanici, Eric Bieber, Mike Schaekermann,
575 Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Mar-
576 cel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke
577 Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni,
578 Nathan Lintz, Tiago Cardal Pais, Henrik Jacobs-
579 son, Idan Szpektor, Nan-Jiang Jiang, and 3290 oth-
580 ers. 2025. [Gemini 2.5: Pushing the frontier with
581 advanced reasoning, multimodality, long context,
582 and next generation agentic capabilities](#). *Preprint*,
583 arXiv:2507.06261.

584 DeepSeek-AI. 2024. [Deepseek-v3 technical report](#).
585 *Preprint*, arXiv:2412.19437.

586 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang,
587 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, Zhi-
hong Shao, Zhuoshu Li, Ziyi Gao, and 181 others.
2025. [Deepseek-r1: Incentivizing reasoning capa-
bility 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 Schel-
ten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh
Goyal, Anthony Hartshorn, Aobo Yang, Archi Mi-
tra, 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 reinforce-
ment learning](#). *Preprint*, arXiv:2510.01932.

Shijue Huang, Hongru Wang, Wanjun Zhong, Zhaochen
Su, Jiazhan Feng, Bowen Cao, and Yi R. Fung.
2025a. [Adactrl: Towards adaptive and controllable
reasoning via difficulty-aware budgeting](#). *Preprint*,
arXiv:2505.18822.

Yuchen Huang, Sijia Li, Zhiyuan Fan, Minghao LIU,
Wei Liu, and Yi R. Fung. 2025b. [Scaling environ-
ments for LLM agents: Fundamentals, approaches,
and future directions](#). In *Workshop on Scaling Envi-
ronments for Agents*.

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. [Re-
wardbench: Evaluating reward models for language
modeling](#). *Preprint*, arXiv:2403.13787.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu,
Fangyu Liu, Julian Eisenschlos, Urvashi Khandel-
wal, 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.

643	Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration . <i>Preprint</i> , arXiv:1802.08802.	700
644		
645		
646		
647	Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning . <i>Preprint</i> , arXiv:2304.08485.	
648		
649		
650	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 . <i>Preprint</i> , arXiv:2504.08942.	
651		
652		
653		
654		
655		
656	Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. 2024. Laser: Llm agent with state-space exploration for web navigation . <i>Preprint</i> , arXiv:2309.08172.	
657		
658		
659		
660	Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation . <i>Preprint</i> , arXiv:2506.01937.	
661		
662		
663		
664		
665	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 . <i>Preprint</i> , arXiv:2506.21252.	
666		
667		
668		
669		
670	Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants . <i>Preprint</i> , arXiv:2311.12983.	
671		
672		
673		
674	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 . <i>Preprint</i> , arXiv:2506.01952.	
675		
676		
677		
678		
679		
680		
681	Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. 2025. Nnetnav: Unsupervised learning of browser agents through environment interaction in the wild . <i>Preprint</i> , arXiv:2410.02907.	
682		
683		
684		
685	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 . <i>Preprint</i> , arXiv:2112.09332.	
686		
687		
688		
689		
690		
691		
692		
693	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 . <i>Preprint</i> , arXiv:2410.21276.	
694		
695		
696		
697		
698		
699		
	OpenAI. 2025. Introducing gpt-5 .	700
	Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024a. Autonomous evaluation and refinement of digital agents . <i>Preprint</i> , arXiv:2404.06474.	701 702 703 704
	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 . <i>Preprint</i> , arXiv:2406.12373.	705 706 707 708 709
	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 . <i>Preprint</i> , arXiv:2411.02337.	710 711 712 713 714 715
	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 . <i>Preprint</i> , arXiv:2403.02502.	716 717 718 719
	Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. Fung. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers . <i>Preprint</i> , arXiv:2506.23918.	720 721 722 723 724 725 726
	Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence . <i>Preprint</i> , arXiv:2507.20534.	727 728 729 730 731 732 733
	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 . <i>Preprint</i> , arXiv:2509.02544.	734 735 736 737 738 739 740 741
	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 . <i>Preprint</i> , arXiv:2505.16421.	742 743 744 745 746 747
	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 . <i>Preprint</i> , arXiv:2501.07572.	748 749 750 751 752
	xAI. 2025. Grok 4 model card .	753

754 Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tian-
755 bao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and
756 Caiming Xiong. 2025. [Aguvis: Unified pure vi-
757 sion agents for autonomous gui interaction](#). *Preprint*,
758 arXiv:2412.04454.

759 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang,
760 Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,
761 Chengen Huang, Chenxu Lv, Chujie Zheng, Day-
762 iheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao
763 Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41
764 others. 2025. [Qwen3 technical report](#). *Preprint*,
765 arXiv:2505.09388.

766 Yuanzhao Zhai, Tingkai Yang, Kele Xu, Feng Dawei,
767 Cheng Yang, Bo Ding, and Huaimin Wang. 2024.
768 [Enhancing decision-making for llm agents via step-
769 level q-value models](#). *Preprint*, arXiv:2409.09345.

770 Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou,
771 Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue
772 Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Gra-
773 ham Neubig. 2024. [Webarena: A realistic web envi-
774 ronment for building autonomous agents](#). *Preprint*,
775 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 Plan Generation

Given the high difficulty of GAIA tasks, which often require specialized domain knowledge to solve, we generate a plan for each question to assist the agent in producing higher-quality trajectories. However, we avoid making the plan overly detailed, so as not to oversimplify the task. Specifically, we first use Gemini-2.5-pro to produce a concise plan based on GAIA’s reference solution, and then refine it manually to ensure that the plan contains no critical intermediate results and is not excessively detailed. The plan includes only suggestive steps, enabling the LLM to perform purposeful exploration. This approach ensures both the quality and the naturalness of the resulting trajectories. The prompts we use, along with an example plan, are shown in Table 6 and Table 7.

A.3 Prompt

This section contains the prompts used for benchmark construction and evaluation, as shown in Tables 8 and 9. Our prompt designs are crafted by human experts, while also drawing upon prior work.

Table 8 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

Prompt for plan generation
<p>You are a web agent responsible for completing the user’s objective.</p> <p>## Your task</p> <p>For the given objective, come up with a simple step by step plan.</p> <p>Do not add any superfluous steps.</p> <p>Make sure that each step has all the information needed - do not skip steps.</p> <p>Do not go into too much detail.</p> <p>## Hint (only for reference)</p> <p>{the gaia solution}</p> <p>### Notes</p> <ul style="list-style-type: none"> - The Hint is provided **only** to help you generate correct plan. - You must **NOT** copy any sentences, phrases, or exact structure from the hint. - You must **NOT** give any **intermediate results**, even in examples. If any **intermediate results** from the hint are required, you must recompute them yourself through independent reasoning.

Table 6: Prompt for plan generation.

Example of Plan
<p>[User objective]</p> <p>If Eliud Kipchoge could maintain his record-making marathon pace indefinitely, how many thousand hours would it take him to run the distance between the Earth and the Moon its closest approach? Please use the minimum perigee value on the Wikipedia page for the Moon when carrying out your calculation. Round your result to the nearest 1000 hours and do not use any comma separators if necessary.</p> <p>[Plan]</p> <ol style="list-style-type: none"> 1. Find Eliud Kipchoge’s world record marathon time and use it to determine his average pace in minutes per mile. 2. Search for the Wikipedia page for the Moon and find the minimum perigee distance in miles. 3. Convert the running pace from minutes per mile to miles per hour. 4. Divide the distance to the Moon by this speed to calculate the total travel time in hours. 5. Round the total time in hours to the nearest thousand and then state the final result in thousands of hours.

Table 7: An example of the generated plans.

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 9, 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.4 Annotation Process

Three PhD-level annotators conduct the annotation, following the instruction presented in Table 10. For each trajectory, they evaluate performance across five dimensions on a 1–4 scale, accompanied by justifications and fine-grained supplementary information as shown in Table 11. All annotators independently label the complete dataset, and their

results are subsequently compared. Discrepancies are resolved through group discussion while samples exhibiting excessive divergence in ratings are excluded from the final dataset.

A.5 Example Trajectory

In Table 12, we provide an example trajectory in which the agent, after attempting to obtain key information, directly draws an incorrect conclusion based on its internal knowledge, representing hallucinatory task progress. Most models, however, markedly overestimate its completeness and truthfulness. Moreover, during the agent’s exploration process, only one redundant step occurred, yet the models significantly underestimated its efficiency. This observation is consistent with the conclusions derived from our error analysis.

A.6 Example Queries from GAIA

Table 13 presents an example for each difficulty level. As the difficulty increases, both the reasoning capability required to solve the problem and the number of steps involved rise substantially.

831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852

853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873

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 8: Prompt for trajectory generation.

A.7 Additional Results

We test a total of 10 models; however, the analysis section does not present the complete results. Here, we provide the remaining results in Table 3 and Figure 4. 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

Your task is to evaluate a web agent's trajectory. The agent need to answer a question by using a browser and other tools. You will receive input in the following format:

...

Question

The question the web agent must answer

Trajectory

Turn 1

Accessibility Tree

Accessibility tree of the current web page

Current URL

Current page URL

Agent Output

The agent's output at this turn

Tool Response

The tool's response at this turn

Turn 2

...

...

You must assign integer scores from 1 to 4 on the following five dimensions, according to the criteria below:

...

Completeness

Definition: How much of the task has been completed, approximated as (correctly executed steps) / (estimated total steps needed).

4: The task is completed and a correct final answer is provided.

3: Progress > 2/3 of the required steps, but the task is not completed.

2: Progress between 1/3 and 2/3 of the required steps.

1: Progress < 1/3 of the required steps.

Adaptability

Definition: The agent's ability to avoid prolonged stagnation or error loops and adjust its strategy.

4: No long-term stagnation occurs.

3: The agent gets stuck, but then adjusts its strategy and significantly advances the task correctly.

2: The agent gets stuck and adjusts strategy, but does not significantly advance the task, or advances it incorrectly.

1: The agent gets stuck, does not adjust strategy, and makes multiple ineffective attempts.

Truthfulness

Definition: Degree of hallucination in the trajectory. Hallucinations include obvious reasoning errors, factual errors, format errors in outputs, and incorrect web interactions.

4: No hallucinations.

3: Minor hallucinations that do not significantly affect task execution, or are quickly corrected.

2: More serious hallucinations that significantly affect task execution but are later corrected.

1: Serious hallucinations that significantly affect task execution and are not corrected.

Efficiency

Definition: Amount of redundant steps. Redundant steps include formatting errors, incorrect web operations, and repeated actions taken in an identical page state.

4: No redundant steps.

3: Redundant steps $\leq 10\%$ of all steps, or at most 1 redundant step.

2: Redundant steps $\leq 20\%$ of all steps, or at most 3 redundant steps.

1: Redundant steps > 20% of all steps.

Soundness

Definition: The agent's ability to obtain key external information to support its answer and reduce reliance on internal knowledge. The agent should gather crucial information (non-common-knowledge obtained from the web) and reason over it.

4: Obtains all key information. If the trajectory stalls during information gathering or yields an incorrect final answer but does not neglect information collection, it is still scored 4.

3: Obtains $\geq 50\%$ of the key information; the remaining information is filled in from internal knowledge.

2: Obtains < 50% of the key information; the rest is inferred from internal knowledge.

1: Produces a final answer based on incomplete information, or relies on hallucinated internal knowledge.

...

You must output ONLY the final scores in the following format, without any explanations or reasoning:

...

Completeness: <score>

Adaptability: <score>

Truthfulness: <score>

Efficiency: <score>

Soundness: <score>

...

Table 9: Prompt for trajectory evaluation.

Instruction given to annotators

You are required to evaluate a given web agent trajectory along five dimensions: Completeness, Adaptability, Truthfulness, Efficiency, and Soundness.

For each dimension:

- Assign a score from 1 to 4, according to the definitions and scoring criteria below.
- Provide a concise justification for your score, including the specific details requested for each dimension.
- Refer to the trajectory step numbers (1, 2, 3...) when specifying certain events or behaviors.

1. Completeness

Definition: Measures how much of the task has been completed, approximated as (correctly executed steps) / (estimated total steps needed).

4: The task is completed and a correct final answer is provided.

3: Progress $> 2/3$ of required steps, but task is not completed.

2: Progress between $1/3$ and $2/3$ of required steps.

1: Progress $< 1/3$ of required steps.

Requirements: Reference the solution provided by GAIA to assess how much of the task the agent completed.

2. Adaptability

Definition: Measures the agent's ability to avoid prolonged stagnation or error loops and to adjust its strategy.

4: No long-term stagnation occurs.

3: The agent gets stuck, but then adjusts its strategy and significantly advances the task correctly.

2: The agent gets stuck and changes strategy, but does not significantly advance the task, or advances it incorrectly.

1: The agent gets stuck, does not adjust strategy, and makes multiple ineffective attempts.

Requirements: If stagnation occurs, specify step number range during which the stagnation happened (e.g., "stagnation between steps 12–20").

3. Truthfulness

Definition: Degree of hallucination in the trajectory. Hallucinations include obvious reasoning errors, factual errors, format errors in outputs, and incorrect web interactions.

4: No hallucinations.

3: Minor hallucinations that do not significantly affect task execution, or are quickly corrected.

2: More serious hallucinations that significantly affect task execution but are later corrected.

1: Serious hallucinations that significantly affect task execution and are not corrected.

Requirements: Identify specific step numbers containing:

- Action Errors (Format errors and incorrect web interactions).
- Hallucinations (including reasoning or factual errors). For each hallucination, briefly explain the nature of the error and its impact on task execution.

4. Efficiency

Definition: Amount of redundant steps. Redundant steps include formatting errors, incorrect web operations, and repeated actions taken in an identical page state.

4: No redundant steps.

3: Redundant steps $\leq 10\%$ of all steps, or at most 1 redundant step.

2: Redundant steps $\leq 20\%$ of all steps, or at most 3 redundant steps.

1: Redundant steps $> 20\%$ of all steps.

Requirements: Identify specific step numbers containing:

- Invalid Action (Format errors and incorrect web interactions).
- Repetitive Action (repeated actions taken in an identical page state).

5. Soundness

Definition: The agent's ability to obtain key external information to support its answer and reduce reliance on internal knowledge. The agent should gather crucial information (non-common-knowledge obtained from the web) and reason over it.

4: Obtains all key information. If the trajectory stalls during information gathering or yields an incorrect final answer but does not neglect information collection, it is still scored 4.

3: Obtains $\geq 50\%$ of the key information; the remaining information is filled in from internal knowledge.

2: Obtains $< 50\%$ of the key information; the rest is inferred from internal knowledge.

1: Produces a final answer based on incomplete information, or relies on hallucinated internal knowledge.

Requirements: Clearly list the key information required to solve the task. List which key information the agent actually obtained during the trajectory.

Table 10: Instruction given to annotators.

Example of annotation result

completeness

- score: 4

- justification: The answer is correct.

adaptability

- score: 3

- justification: progress stalled, after adjusting strategy can significantly advance the task.

- stagnation: 5-26

truthfulness

- score: 4, - justification: No obvious hallucinations or errors.

- hallucination:

- action error:

efficiency

- score: 1

- justification: 8 redundant steps exists

- invalid action:

- repetitive action:

soundness

- score: 3

- justification": "Obtained 50% of key information, used internal knowledge to obtain the two-time winners of "Survivor"

- key information: the number of unique winners of "Survivor" up to season 44, the number of winners of "American Idol"

- obtained information: the number of winners of "American Idol"

Table 11: Example of annotation result.

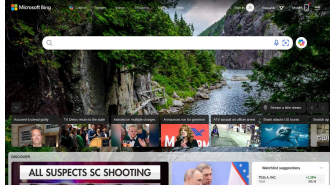
Example Trajectory

[Question]

In Unlambda, what exact character or text needs to be added to correct the following code to output "For penguins"? If what is needed is a character, answer with the name of the character. If there are different names for the character, use the shortest. The text location is not needed. Code:

```
'r'''''''.F.o.r .p.e.n.g.u.i.n.si
```

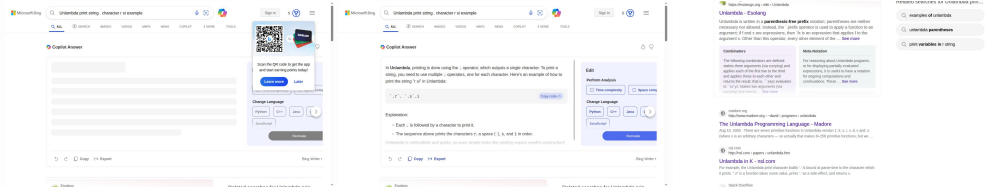
[Turn 1]



[action]

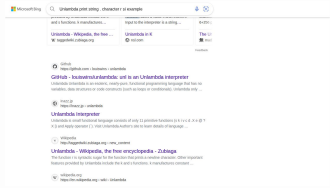
type("Unlambda print string . character r si example", element_id=4)

[Turn 2-4]



...

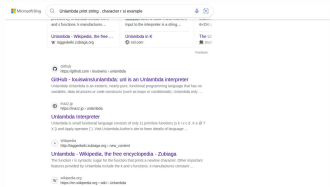
[Turn 7]



[action]

click(9627)

[Turn 8]



[action]

click(9627)

[Turn 9]

[Thought]

The provided Unlambda code prints each character using .x, but the final 's' is missing the leading '.' and is written as 'si' (which are combinators). To print the letter 's', a '.' is needed before 's'. Therefore, the missing character is "dot".

[Action]

commit_answer("dot")

[Correct Answer]

backtick

[Score]

completeness: 1, adaptability: 2, truthfulness: 1, efficiency: 3, soundness: 2

[GPT-5 Result]

completeness: 4, adaptability: 2, truthfulness: 4, efficiency: 2, soundness: 3

[Grok-4 Result]

completeness: 4, adaptability: 1, truthfulness: 3, efficiency: 1, soundness: 4

Table 12: Example Trajectory

Example Queries

[Level 1]

If Eliud Kipchoge could maintain his record-making marathon pace indefinitely, how many thousand hours would it take him to run the distance between the Earth and the Moon its closest approach? Please use the minimum perigee value on the Wikipedia page for the Moon when carrying out your calculation. Round your result to the nearest 1000 hours and do not use any comma separators if necessary.

[Level 2]

What are the EC numbers of the two most commonly used chemicals for the virus testing method in the paper about SPFMV and SPCSV in the Pearl Of Africa from 2016? Return the semicolon-separated numbers in the order of the alphabetized chemicals.

[Level 3]

I was referencing each of the tables in the file from papers that were cited by the "Trans fatty acid contents in chocolates and chocolate wafers in Turkey" paper. I lost my own reference sheet and need to know which of the papers each table came from. The file may not use the full table caption. If the references in the "Trans fatty acid" paper bibliography were numbered starting with 1, give me the numbers in the order that they would be used to fill the cells in the Excel file from top to bottom, as a comma separated list.

To solve the task above, you will have to use this attached file: 3da89939-209c-4086-8520-7eb734e6b4ef.xlsx

Table 13: Example Queries

Model	Level	Comp	Adapt	Truth	Eff	Sound	All	Exact	Rank
GPT-5	1	67.31	61.54	55.77	61.54	63.46	61.92	28.85	73.08
	2	59.65	58.77	40.35	48.25	46.49	50.70	14.04	64.91
	3	55.88	44.12	32.35	41.18	44.12	43.53	2.94	76.47
GPT-5-mini	1	63.46	61.54	53.85	57.69	57.69	58.85	26.92	76.92
	2	51.75	53.51	38.60	47.37	57.02	49.65	15.79	57.89
	3	41.18	35.29	29.41	44.12	41.18	38.24	5.88	70.59
Gemini-2.5-pro	1	61.54	65.38	61.54	55.77	90.38	66.92	25.00	80.77
	2	61.40	63.16	42.11	49.12	72.81	57.72	14.04	71.93
	3	55.88	50.00	38.24	29.41	58.82	46.47	2.94	58.82
Gemini-2.5-flash	1	63.46	78.85	46.15	69.23	71.15	65.77	32.69	65.38
	2	60.53	65.79	41.23	48.25	67.54	56.67	14.91	63.16
	3	55.88	52.94	38.24	44.12	47.06	47.65	0.00	52.94
DeepSeek-V3.2	1	65.38	67.31	42.31	57.69	65.38	59.62	25.00	61.54
	2	62.28	54.39	35.96	47.37	52.63	50.53	12.28	75.44
	3	47.06	50.00	38.24	35.29	44.12	42.94	5.88	64.71
Grok-4	1	63.46	55.77	57.69	51.92	75.00	60.77	25.00	84.62
	2	62.28	54.39	44.74	46.49	64.91	54.56	14.91	71.93
	3	52.94	41.18	44.12	32.35	47.06	43.53	0.00	76.47
Claude-sonnet-4.5	1	65.38	71.15	50.00	51.92	53.85	58.46	21.15	61.54
	2	59.65	64.91	47.37	43.86	44.74	52.11	14.04	71.93
	3	52.94	64.71	41.18	38.24	35.29	46.47	0.00	76.47
Kimi-K2	1	67.31	67.31	53.85	63.46	63.46	63.08	30.77	46.15
	2	58.77	55.26	42.11	47.37	66.67	54.04	17.54	45.61
	3	58.82	50.00	38.24	50.00	41.18	47.65	5.88	58.82
Qwen3-30b-a3b	1	63.46	73.08	44.23	65.38	61.54	61.54	30.77	38.46
	2	60.53	60.53	35.09	51.75	61.40	53.86	18.42	26.32
	3	50.00	44.12	35.29	47.06	38.24	42.94	5.88	35.29
Llama3.3-8b	1	59.62	65.38	53.85	53.85	71.15	60.77	26.92	26.92
	2	57.89	57.02	37.72	44.74	72.81	54.04	17.54	38.60
	3	38.24	41.18	29.41	32.35	50.00	38.24	2.94	29.41

Table 14: 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
GPT-5	chosen	70.00	61.00	58.00	53.00	61.00	60.60	32.00
	rejected	52.00	53.00	28.00	48.00	40.00	44.20	0.00
GPT-5-mini	chosen	71.00	66.00	64.00	56.00	66.00	64.60	34.00
	rejected	35.00	39.00	18.00	43.00	43.00	35.60	0.00
Gemini-2.5-pro	chosen	75.00	68.00	61.00	48.00	75.00	65.40	25.00
	rejected	46.00	55.00	32.00	47.00	75.00	51.00	5.00
Gemini-2.5-flash	chosen	75.00	75.00	64.00	54.00	71.00	67.80	33.00
	rejected	46.00	59.00	20.00	52.00	59.00	47.20	1.00
DeepSeek-V3.2	chosen	76.00	69.00	53.00	50.00	67.00	63.00	29.00
	rejected	45.00	45.00	23.00	46.00	42.00	40.20	0.00
Grok-4	chosen	75.00	55.00	62.00	41.00	70.00	60.60	23.00
	rejected	47.00	50.00	34.00	50.00	59.00	48.00	7.00
Claude-sonnet-4.5	chosen	73.00	73.00	58.00	46.00	59.00	61.80	26.00
	rejected	47.00	60.00	36.00	44.00	32.00	43.80	1.00
Kimi-K2	chosen	75.00	73.00	66.00	64.00	70.00	69.60	38.00
	rejected	47.00	42.00	23.00	40.00	53.00	41.00	0.00
Qwen3-30b-a3b	chosen	76.00	70.00	61.00	64.00	71.00	68.40	39.00
	rejected	43.00	52.00	14.00	45.00	44.00	39.60	0.00
Llama3.3-8b	chosen	72.00	68.00	66.00	60.00	74.00	68.00	35.00
	rejected	38.00	45.00	15.00	30.00	63.00	38.20	0.00

Table 15: 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.