WebArXiv: A Reproducible Benchmark for Evaluating Multimodal Web Agents on arXiv Tasks

Anonymous ACL submission

Abstract

Recent advances in foundation models have enabled autonomous web agents to navigate and interact with real-world websites. However, existing benchmarks primarily focus on general-purpose web navigation, offering limited coverage of information-centric environments. Evaluations that depend on live websites further hinder reproducibility due to constantly changing content. The arXiv platform provides a natural balance between realism and reproducibility, featuring hierarchically structured and information-centric webpages without privacy-sensitive interactions. Building on this foundation, we present WebArxiv, a benchmark for reproducible evaluation of multimodal web agents within the arXiv environment. WebArxiv is built from static webpage snapshots and includes 510 time-invariant tasks, each with a unique and deterministic ground truth. We evaluate a range of foundation-model-based web agents, revealing that WebArxiv poses significant challenges for current web agents. Behavioral analyses reveal a common failure mode in which agents over-rely on fixed interaction histories, leading to incomplete or repetitive reasoning. To address this limitation, we equip the agents with a lightweight dynamic memory mechanism that enables adaptive retrieval and reasoning over relevant context, thereby enhancing their overall navigation performance.

1 Introduction

The rapid advancement of large language and multimodal models has led to the emergence of autonomous web agents capable of performing complex tasks on real-world websites (Garg et al., 2025; Yang et al., 2025). However, most existing benchmarks focus on commercial or general-purpose web navigation, while information-centric scientific platforms remain underexplored (Yehudai et al., 2025). Such platforms demand reasoning over structured metadata and hierarchical taxonomies, and are essential for evaluating agents supporting scholarly workflows.

The arXiv platform serves as an ideal testbed, as it provides a more stable and transparent environment than comparable scholarly platforms, with well-structured metadata and minimal personalization. To our knowledge, WebVoyager (He et al., 2024) is the only existing benchmark that incorporates arXiv as an evaluation environment. However, it includes only a few tasks and is largely confined to basic web navigation, falling short of capturing scholarly workflows. Moreover, because WebVoyager operates on live webpages, its task answers change over time. In response, WebVoyager adopts an automatic evaluation protocol based on GPT-4V, but this reliance on model-based judgment introduces additional inconsistency. These limitations highlight the need for reproducible evaluation of agents in scientific environments.

To address this need, we introduce WebArxiv, a benchmark that enables reproducible evaluation of multimodal web agents within the arXiv environment. Within WebArXiv, we define a diverse set of realistic scholarly tasks that go beyond basic information queries and rule compliance, emphasizing multi-condition paper retrieval, deep content extraction, and cross-paper comparison, capabilities that have been largely underexplored in prior benchmarks. To guarantee reproducibility, all tasks are built on static snapshots of the arXiv website, forming a set of time-invariant evaluations. Each task is paired with reference action trajectories and deterministic ground truths, enabling consistent and verifiable comparisons across diverse web agents.

We conduct baseline experiments with ten representative agent configurations on the WebArxiv benchmark, including general-purpose LMM-based web agents such as GPT-4o (OpenAI, 2024) and several specialized web agents like OpenWebAgent (Iong et al., 2024b). The results

show that WebArxiv is challenging for current web agents. Even the strongest model, GPT-o1, achieves only a 54.1% overall success rate, followed by Gemini-2.5 at 53.7%, while most other models remain below 45%. In particular, most models struggle with advanced search tasks, such as multi-condition paper retrieval and paper comparison.

Further analysis reveals a common failure mode among web agents: due to the complexity of web states, existing agents rely on only a very short recent interaction history, which often leads to repeated error patterns. To address this issue, we enable web agents to adaptively retrieve the most relevant prior interactions. This design not only helps prevent agents from entering error loops, but also enables more efficient task retries by letting agents effectively reuse previously acquired information when recovering from task failures (e.g., being redirected to the homepage.)

Experimental results show that this mechanism yields consistent performance improvements across nearly all evaluated models.

Our contributions are summarized as follows:

- We introduce WebArxiv, a benchmark that enables reproducible evaluation of multimodal web agents within the hierarchically structured arXiv environment.
- We conduct a comprehensive evaluation of ten advanced web agents on WebArxiv, demonstrating clear baseline performance across different task types and query complexity levels.
- Through detailed analyses, we find that existing web agents tend to over-rely on fixed interaction histories. To address this issue, we introduce a lightweight dynamic memory mechanism that allows web agents to adaptively leverage past interactions.

2 Related Work

Benchmarks for Web Agent. With the rapid development of large language model (LLM) and large multimodal model (LMM) based web agents, a series of benchmarks have been proposed to evaluate their capabilities (Liu et al., 2023). Early efforts focused on simulated commercial environments. For example, WebShop (Yao et al., 2022a) simulates a virtual online store populated with more than one million product listings scraped from Amazon. Later, WebArena (Zhou et al., 2023) extends evaluation to multi-domain web environments, simulating websites spanning e-commerce, social media, and productivity to improve task diversity.

More recently, several benchmarks have begun evaluating agents directly on the live Web rather than in simulated environments. For instance, Mind2Web (Deng et al., 2023) collects large-scale trajectories of real web interactions across thousands of sites, while BrowseComp (Wei et al., 2025) emphasizes complex comparative reasoning and open-domain browsing behaviors. WebVoyager (He et al., 2024) further extends this line of work to multimodal web agent evaluation, covering tasks across 15 real-world websites that integrate textual and visual information. These benchmarks capture the unpredictability of real-world web navigation(Drouin et al., 2024). However, a major limitation of live environments is answer drift, as webpage content evolves over time (de Chezelles et al., 2025). This dynamic nature reduces reproducibility, necessitates frequent updates to gold-standard answers. In response, WebVoyager (He et al., 2024) and Mind2Web 2 (Gou et al., 2025a) adopt model-as-a-judge protocols to automatically assess task outcomes and thereby reduce human annotation overhead. Overall, existing benchmarks have advanced the evaluation of web agents in general-purpose domains, yet they provide limited coverage of information-centric workflows (Wang et al., 2022). This gap motivates the development of new benchmarks that preserve real-world complexity while ensuring reproducibility and stable ground truths (Levy et al., 2025).

Web Agents. Recent advances in LLMs and LMMs have enabled web agents to complete end-to-end user instructions through direct interaction with real-world websites (Ma et al., 2024). ReAct (Yao et al., 2022b) introduced interleaved reasoning and acting, prompting models to generate both intermediate reasoning traces and executable action sequences. Reflexion (Shinn et al., 2023) and Auto-Eval and Refine (Pan et al., 2024a) further allow agents to analyze past failures and iteratively refine their decision strategies. WebPilot (Zhang et al., 2025b) extends this approach by incorporating Monte Carlo Tree Search (MCTS), enabling agents to evaluate, backtrack, and optimize their decision trajectories more effectively (Koh et al., 2024). Furthermore, fine-tuning language or multimodal models has emerged as another effective strategy to enhance agents' capabilities on web tasks (Pan et al., 2024b). Representative examples include text-finetuned agents such as WebGPT

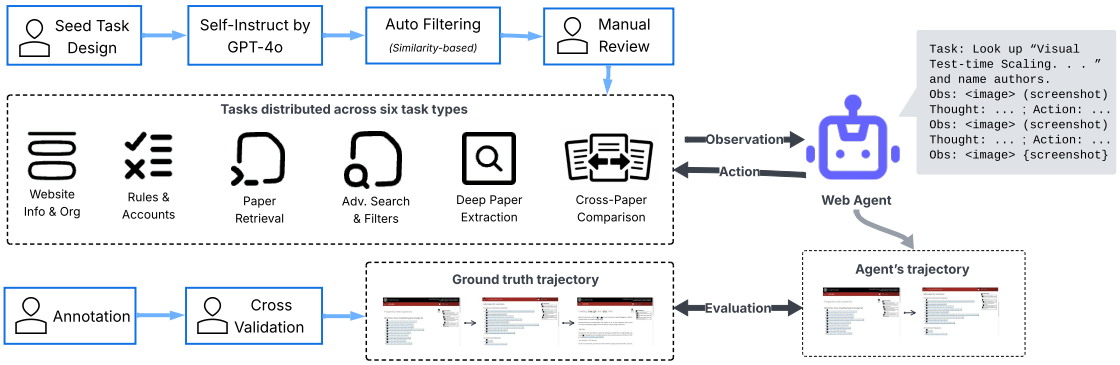


Figure 1: WebArXiv benchmark creation pipeline, showing seed task design, self-instruct generation by GPT-4o, automatic similarity-based filtering, and manual review. The resulting tasks are distributed across six categories and annotated through expert cross-validation to form ground-truth trajectories for web-agent evaluation.

(Nakano et al., 2023), which fine-tunes GPT models for web browsing and question answering, and HTML-pretrained agents such as WebAgent (Iong et al., 2024a), which leverage large-scale HTML corpora to improve structural understanding and action grounding (Hong et al., 2024).

In contrast to LLM-based agents that rely primarily on DOM or HTML parsing, LMM-based agents perceive webpages through visual modalities, enabling a more holistic understanding of layout and content (Gou et al., 2025b). Most existing multimodal agents are built on closed-source LMMs, such as GPT-4V (Zheng et al., 2024a) and Gemini (Georgiev et al., 2023), which exhibit strong visual grounding and reasoning capabilities. These models facilitate effective action planning through paradigms such as ReAct (Yao et al., 2022b) and Reflexion (Shinn et al., 2023). Representative multimodal web-agent frameworks, including Pix2Act (Shaw et al., 2023) and WebGUM (Furuta et al., 2024), operate directly on webpage screenshots, translating visual cues into grounded actions without explicit reliance on DOM structures. SeeAct (Zheng et al., 2024b) further extends this paradigm by integrating visual grounding with tool-augmented candidate selection, enabling more precise element localization and robust interaction reasoning.

3 WebArxiv Benchmark

3.1 Overview

We introduce WebArXiv, a reproducible benchmark for evaluating web agents in the arXiv platform. WebArXiv comprises 510 tasks that emphasize complex scholarly web operations, including multi-condition paper retrieval, deep content extraction, and cross-paper comparison. Each task is packaged with a natural-language instruction, a reference action trajectory, and a machine-verifiable ground truth validated by multiple annotators. Figure 1 provides an overview of the benchmark creation pipeline. This section details the problem formulation, task taxonomy and statistics, task construction pipeline, and annotation process.

3.2 Problem Formulation

Following prior work (Zhou et al., 2024), we formalize the web environment as a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998), represented by the tuple $\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O})$, where $\mathcal{S}$ denotes the underlying state space of the web interface; $\mathcal{A}$ is the action space, consisting of discrete user-interaction primitives such as click and scroll; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ defines a deterministic transition function that maps the current page state and executed action to a new web state after rendering (Sutton and Barto, 2018); and $\mathcal{O}$ represents the observation space, comprising multimodal inputs available to the agent, including the rendered webpage screenshot and its corresponding textual elements.

Given a task instruction i, the current multimodal observation $o_t \in \mathcal{O}$, and the interaction history $a_{1:t-1}$, the agent selects an action $a_t \in \mathcal{A}$.

Upon execution, the environment transitions deterministically to a new state $s_{t+1} = \mathcal{T}(s_t, a_t)$, and the agent receives an updated observation $o_{t+1} \in \mathcal{O}$, which reflects the new visual and textual interface after the action is applied.

3.3 Task Types

WebArxiv consists of 510 tasks grouped into six categories, each targeting a distinct aspect of scholarly web interaction:

- **Website Information and Organizational Details.** This category evaluates agents' ability to navigate the arXiv interface and locate general

Statistic / Category	Website Info & Org	Rules & Accounts	Paper Retrieval	Adv. Search & Filters	Deep Paper Extraction	Cross-Paper Comparison	Total
Generated tasks	100	100	200	200	200	200	1000
Similar tasks	34	37	31	51	28	44	225
Unable to annotate	5	5	7	11	32	38	98
Expired tasks	6	3	62	38	40	18	167
Refined tasks	55	55	100	100	100	100	**510**
GT Annotate steps	270	274	639	752	884	1613	4432
Avg. GT steps/task	4.9	5.0	6.4	7.5	8.8	16.1	8.69

Table 1: Data distribution across six task categories. GT: Ground Truth

information such as editorial structures, website information and organizational guidelines.

- **Rules, Licensing, and User Account Management.** These tasks assess whether agents can identify procedural and policy-related content, including submission requirements, license types, withdrawal policies, and user account operations.
- **Research Paper Discovery and Retrieval.** This category measures agents' capability to search for scholarly papers by author, ID, or category, while extracting stable metadata such as titles, authors, and primary subject areas.
- **Advanced Search and Filtering.** Tasks in this category involve executing complex multi-condition queries by combining keywords, publication years, and subject domains to evaluate multi-constraint reasoning and structured search proficiency.
- **Deep Paper Content Extraction.** These tasks require extracting fine-grained information directly from the HTML version of papers, testing agents' ability to identify detailed textual and structural content.
- **Cross-Paper Comparison.** These tasks require comparing and synthesizing information across multiple papers, evaluating agents' ability to perform cross-document reasoning over sections, experiments, and reported findings.

The 510 tasks are distributed across the six categories. Among them, the Advanced Search and Filtering, Deep Paper Content Extraction and Cross-Paper Comparison categories involve the highest number of interaction steps. Detailed task statistics are provided in Table 1, and task examples are presented in Appendix A.1.

3.4 Task Construction Pipeline

Human experts first designed a small set of 20 seed tasks for each category, serving as exemplars for LLM-assisted augmentation. Guided by predefined category schemas and manually crafted templates, GPT-4o was employed to generate 100 candidate tasks for simple categories, and 200 for complex categories, simulating realistic user queries and interaction intents.

The dataset was then refined through a two-stage filtering process. In the first stage, we computed pairwise sentence-level similarities using the `all-mpnet-base-v2` embedding model and removed semantically redundant tasks, retaining only those with similarity scores below 0.8 to ensure diversity of task intent. In the second stage, the remaining tasks were manually reviewed by the research team to confirm their clarity. Specifically, we excluded tasks that could not be annotated reliably, had non-unique ground truths, or were not time-invariant. After this filtering process, the finalized tasks were incorporated into the WebArxiv benchmark for reproducible evaluation. The distribution of tasks across categories is summarized in Table 1.

3.5 Benchmark Construction Details

To ensure reproducibility and deterministic evaluation, we adopt a controlled benchmark construction pipeline covering snapshotting, rendering, and annotation.

Snapshotting and Tools. We used Selenium WebDriver (Chrome v118.0) on Ubuntu 22.04 LTS to capture full-page HTML and screenshots at a fixed 1920×1080 viewport.

Depth and Caching. Each URL was crawled from the root arXiv category page, caching static HTML, linked stylesheets, and embedded scripts locally.

Deterministic Rendering Each snapshot was re-rendered in a sandboxed browser using the same engine to ensure pixel-consistent replay.

Dynamic Content Handling. As arXiv's HTML is server-generated and not JavaScript-dependent, all pages remain stable under caching.

Reproducibility Assurance. Each task directory includes the raw HTML snapshot, hashed metadata, and trajectory JSON file, guaranteeing identical replay across systems.

3.6 Annotation Process

We recruited five PhD-level annotators with experience using the arXiv platform to manually annotate all 510 tasks. For each task category, a reference example was provided to guide annotation consistency. Each annotator independently interacted with the WebArxiv interface to complete their assigned tasks. During this process, they documented every action step and captured key screenshots to record the reasoning process leading to the final answer. Reference trajectories were produced using a custom Playwright v1.45 interface; each click, scroll, and input action was logged to ensure full traceability.

After completing the individual annotations, the annotators performed cross-validation for each task to ensure correctness, consistency, and agreement. The resulting consensus trajectory then served as the ground truth for evaluation.

To assess web agent performance, we compared each model's final output against these ground truth. In addition to final-answer matching, we also examined the agents' full action trajectories via screenshots to evaluate the logical consistency of their reasoning process. Task outcomes were labeled as: **Correct**: output matches the gold-standard answer; **Incorrect**: the agent failed to retrieve the required content; **Partially Correct**: the agent's trajectory was close to completion but missed the final steps. We label a task as Partially Correct if the executed interaction trajectory overlaps with the reference trajectory by more than 70%.

3.7 Dynamic Memory.

In baseline evaluations on WebArXiv, agents often over-rely on the most recent truncated view and overlook earlier task-relevant information, leading to reasoning loops or incomplete answers. These failure patterns point to limitations of rigid memory designs used by most existing web agents.

Based on these observations, we equip agents within WebArXiv with a lightweight dynamic memory mechanism. At each interaction step t, the agent retrieves from the accumulated interaction history $\mathcal{H}_t = \{(o_1, a_1), (o_2, a_2), \ldots, (o_t, a_t)\}$. Instead of selecting fixed trace-back steps, the agent adaptively identifies a *set* of relevant past indices $J^* \subseteq \{1, \ldots, t\}, \quad |J^*| \leq m$, based on its internal assessment of which past context is useful for the current decision. The retrieved memory is then formed as $\mathcal{M}_t = \{o_j \mid j \in J^*\}$, and combined with the current view o_{t+1} to construct the contextual input for action generation. Concretely, the next action is produced by conditioning on the task instruction, current view, and the retrieved memory set: $a_{t+1} \sim \pi\big(i,\, o_{t+1},\, \mathcal{M}_t\big)$. The resulting action is executed in the environment, and the interaction history is updated accordingly. The proposed dynamic memory provides two practical benefits observed in our analysis: It reduces repetitive reasoning loops by reusing earlier task-relevant context, and enables more efficient task retries by allowing agents to reuse previously acquired information when recovering from failures (e.g., after being redirected to the homepage).

4 Experiments

4.1 Baselines

We evaluate a set of mainstream web agents under an end-to-end setting on the WebArxiv benchmark. The selected agents include both general-purpose LMMs that operate in an instruction-following manner and specialized web agents designed for fine-grained browser interaction.

General-purpose LMMs. These models receive natural language instructions along with webpage screenshots as input and generate textual reasoning and action predictions. The temperature parameter for all LMMs was set to 0.1.

- **GPT-o1:** A state-of-the-art multimodal model developed by OpenAI that accepts both image and text input. We provide webpage screenshots and task instructions as input.
- **GPT-4-Turbo:** A high-efficiency variant of GPT-4 with comparable reasoning capabilities but optimized for inference latency.
- **Gemini Series** (DeepMind, 2024): Google DeepMind's multimodal family supporting vision-language understanding. We test Gemini 1.5 pro, 2.0, and 2.5 under the same prompting configuration as GPT-4o, combining textual instructions with webpage screenshots.
- **GPT-4o-mini / GPT-4o:** Compact and full-sized versions of the GPT-4o model family, used to examine the trade-off between model size and web task performance.

Specialized Web Agents. These agents are explicitly designed for browser control. They typ-

Web Agents	Platform & Org Info	Rules & Accounts	Paper Retrieval	Adv. Search & Filters	Deep Paper Extraction	Cross-Paper Comparison	Total (%)
GPT-4-Turbo	43.6%	34.5%	47.3%	25.8%	30.9%	14.8%	36.4%
GPT-4o	36.1%	29.6%	34.5%	25.7%	38.2%	18.5%	32.3%
GPT-o1	**72.7%**	50.3%	**65.5%**	43.2%	44.5%	20.7%	**54.1%**
GPT-o4-mini	52.7%	48.2%	56.4%	29.1%	32.7%	15.0%	43.8%
Gemini-1.5-pro	47.3%	42.2%	52.7%	34.0%	37.8%	19.2%	42.9%
Gemini-2.0	34.5%	29.1%	34.8%	25.2%	27.3%	11.4%	30.6%
Gemini-2.5	65.2%	**57.3%**	52.7%	**47.3%**	35.4%	**21.6%**	53.7%
SeeAct	28.2%	20.0%	25.7%	20.8%	24.9%	9.1%	23.6%
LiteWebAgent	43.7%	47.3%	43.4%	32.3%	**45.5%**	19.9%	44.0%
OpenWebAgent	34.5%	38.9%	43.6%	34.5%	18.2%	8.4%	33.8%

Table 2: Performance comparison of web agents across six task categories in the WebArxiv benchmark.

Web Agents	Platform & Org Info	Rules & Accounts	Paper Retrieval	Adv. Search & Filters	Deep Paper Extraction	Cross-Paper Comparison	Total (%)
GPT-4-Turbo	43.6%	34.5%	47.3%	25.8%	30.9%	14.8%	36.4%
GPT-4-Turbo + dynamic memory	52.6%	42.7%	46.4%	30.0%	29.1%	14.7%	40.2%
GPT-4o	36.1%	29.6%	34.5%	25.7%	38.2%	18.5%	32.7%
GPT-4o + dynamic memory	63.6%	60.0%	38.2%	34.5%	52.7%	25.9%	38.4%
GPT-o1	72.7%	50.3%	65.5%	43.2%	44.5%	20.7%	54.1%
GPT-o1 + dynamic memory	73.3%	55.5%	64.5%	52.7%	60.2%	24.6%	61.8%
GPT-o4-mini	52.7%	48.2%	56.4%	29.1%	32.7%	15.0%	43.8%
GPT-o4-mini + dynamic memory	57.3%	31.8%	52.7%	30.9%	35.5%	17.1%	41.6%
Gemini-1.5-pro	47.3%	42.2%	52.7%	34.0%	37.8%	19.2%	42.9%
Gemini-1.5-pro + dynamic memory	59.7%	59.1%	51.8%	38.2%	45.5%	24.3%	50.9%
Gemini-2.5	65.2%	57.3%	52.7%	47.3%	35.4%	21.6%	53.7%
Gemini-2.5 + dynamic memory	81.8%	72.7%	56.4%	43.6%	41.1%	28.8%	60.0%

Table 3: Comparison of base models and their dynamic memory enhanced models across six task categories.

ically rely on DOM parsing, fine-grained action spaces (e.g., click, type), and internal state tracking for reasoning.

- **SeeAct** (Zheng et al., 2023): A vision-based web agent that integrates a perception module (CLIP) with an action decoder. It employs a global planning strategy and performs step-wise interactions using webpage screenshots.
- **LiteWebAgent** (Zhang et al., 2025a): A lightweight web automation framework that parses DOM structures and uses language models to predict high-level actions. It is optimized for both speed and interpretability.
- **OpenWebAgent** (Iong et al., 2024b): A modular web agent architecture featuring DOM-based environment modeling, visual grounding, and tool-use capabilities. It supports retrieval-augmented inputs and maintains an explicit memory of previous steps.

4.2 Evaluation Protocol

We adopt task success rate as the primary evaluation metric, which measures the proportion of tasks the agent retrieves the correct final answer. Each agent is evaluated on all tasks in the WebArxiv benchmark, and success is determined by comparing the agent's final response with the verified gold-standard answer. The evaluation is conducted under a strict matching criterion to ensure answer accuracy. We performed each task three times and report the averaged results for ten web agents across six task categories in the WebArxiv benchmark.

4.3 Main Results

Table 2 summarizes performance across six task categories and shows clear model specialization. Among base models, GPT-o1 achieves the best overall success rate (54.1%), leading on Platform & Org Info (72.7%) and Paper Retrieval (65.5%).

Gemini-2.5 performs best on Rules & Accounts (57.3%) and Advanced Search & Filters (47.3%), while LiteWebAgent attains the highest score on Deep Paper Extraction (45.5%). Cross-Paper Comparison is the lowest-performing category overall: the strongest base results are Gemini-2.5 (21.6%) and GPT-o1 (20.7%), indicating that cross-document comparison remains challenging for current web agents.

Table 3 compares base models with their dynamic memory variants. Dynamic memory improves performance for 5 of the 6 evaluated models, with GPT-o1 + dynamic memory achieving the best overall success rate (61.8%) and Gemini-2.5 + dynamic memory reaching 60.0% (up from 53.7%). Notably, dynamic memory yields consistent gains across categories and also improves Cross-Paper Comparison in relative terms, suggesting that adaptive context retrieval is an effective and lightweight enhancement for web-agent performance.

4.4 Ablation Studies

We further conducted additional experiments on GPT-4 and GPT-o1 to compare the effectiveness of dynamic memory against fixed-window memory configurations. As shown in Table 4, the dynamic memory variant of GPT-o1 achieved a success rate of 61.8%, outperforming simpler baselines that used only the most recent step (60.0%) or a uniform three-step memory (54.1%). Similarly, introducing dynamic memory improved GPT-4-Turbo from 36.4% to 40.2%, demonstrating its effectiveness in adaptive context retrieval and action generation.

4.5 Analysis

To gain deeper insights into agent behavior, we analyze performance across categories, models, and memory settings, and further investigate typical error patterns observed in the WebArxiv benchmark.

Cross-Category Performance. The distribution of results across categories reveals a clear and consistent gradient of difficulty. Figures 8 and 9 present the performance of GPT-o1 across six task types. Platform & Org Info and Paper Retrieval are comparatively straightforward, with several agents surpassing 60% accuracy. These tasks typically involve single-page lookups or structured metadata extraction, requiring minimal multi-step planning. Rules & Accounts and Deep Paper Extraction are of intermediate difficulty. The former is challenging because tasks often demand retrieval of exact policy statements, where even slight paraphrasing is penalized under strict evaluation criteria. The latter requires navigation of long documents and reliable handling of HTML anchors, such as identifying captions or figures embedded deep within papers.

Advanced Search & Filters presents a higher level of difficulty, as tasks require precise coordination of multiple constraints, including Boolean operators, date ranges, and field-specific filters. Even the strongest base model, Gemini-2.5, achieves only 47.3% accuracy in this category, indicating the fragility of long, constraint-sensitive interaction sequences.

Cross-Paper Comparison emerges as the most challenging category overall. Across all agents, success rates are substantially lower than in any other task type, with even the best-performing models remaining near or below the 25% mark. This reflects the compounded difficulty of retrieving, aligning, and jointly reasoning over information distributed across multiple documents, where errors accumulate from inconsistent cross-referencing or failure to maintain a coherent multi-paper reasoning state.

Cross-Model Comparison. Across models, performance varies substantially. GPT-o1 achieves the best overall score at 54.1%, showing particular strength in Platform and Retrieval tasks. Gemini-2.5, in contrast, leads in Rules & Accounts with 57.3% and in Advanced Search with 47.3%, suggesting that it is more effective at parsing long-form documentation and reasoning over structured form inputs. LiteWebAgent obtains the highest score in Deep Paper Extraction with 45.5%, reflecting the benefit of DOM parsing for anchor-dense HTML content. By comparison, SeeAct consistently underperforms across all categories, illustrating the limitations of purely screenshot-based approaches in text-heavy academic interfaces. Model scale, however, is not a decisive factor; smaller models such as GPT-4o-mini still achieve competitive performance, reaching 43.8% overall. The superior results of GPT-o1 may also be attributed in part to its stronger chain-of-thought reasoning capability.

Dynamic Memory Effects. Introducing dynamic memory yields consistent improvements for most LLM-driven agents, as shown in Table 3. The most notable gains are observed in Deep Paper Extraction, where GPT-o1 rises from 44.5% to 60.2% and GPT-4o from 38.2% to 52.7%. Dynamic memory

Memory Mechanism	Successful (↑)	Partial (↓)	Failed (↓)
GPT-4-Turbo last 3 steps	36.4%	18.2%	45.5%
GPT-4-Turbo last 2 steps	34.5%	20.4%	45.2%
GPT-4-Turbo last step	**43.6%**	**14.5%**	**41.8%**
GPT-4-Turbo + dynamic memory	40.2%	16.3%	43.6%
GPT-o1 last 3 steps	54.1%	16.2%	27.0%
GPT-o1 last 2 steps	58.2%	15.1%	26.8%
GPT-o1 last step	60.0%	14.4%	25.7%
GPT-o1 + dynamic memory	**61.8%**	**12.7%**	**25.5%**

Table 4: Task success rates of GPT-o1 and GPT-4 Turbo models under different memory strategies. The baseline uses the last 3 steps to make decisions, while dynamic memory only use the most relevant step to make decisions.

also benefits Platform & Org Info and Rules & Accounts tasks by enabling agents to retrieve relevant earlier views instead of relying on fixed-length histories.

For Advanced Search & Filters, the effects are mixed: GPT-o1 improves from 43.2% to 52.7%, while GPT-4-Turbo shows only marginal gains. In Cross-Paper Comparison, dynamic memory yields clear improvements despite low absolute performance, with Gemini-2.5 increasing from 21.6% to 28.8% and GPT-4o from 18.5% to 25.9%. This indicates that adaptive context retrieval is particularly beneficial for cross-document reasoning.

The one exception is GPT-o4-mini, which experiences a slight performance decline, likely due to over-reliance on misleading prior states given its reduced reasoning depth.

Error Analysis. Qualitative inspection of trajectories reveals recurring errors. Agents often fixate on recent but irrelevant history, showing rigid attention patterns. Strict string-matching exposes brittleness, penalizing semantically correct yet paraphrased answers, especially in policy tasks. Errors in field and anchor selection also occur on advanced search or HTML-dense pages, where agents confuse fields (e.g., title vs. abstract) or fail to activate the correct checkbox or link.

These findings carry several implications for the design of future web agents. First, incorporating lightweight DOM priors can mitigate failures on anchor-dense pages. Second, memory mechanisms should be selective rather than uniformly long, with dynamic memory offering a lightweight but effective solution. Third, constraining action templates for structured forms reduces drift when managing Boolean operators or field assignments.

5 Conclusion

We introduced WebArxiv, a reproducible benchmark designed for evaluating web agents within the arXiv platform. WebArxiv consists of time-invariant tasks that capture realistic scholarly interaction scenarios. Using this benchmark, we conducted extensive experiments to evaluate both general-purpose LMMs and specialized web agents. Empirical observations revealed key behavioral limitations, such as over-reliance on fixed interaction histories. In response, we equip the agent with a lightweight dynamic memory mechanism that enables it to adaptively retrieve and reason over relevant context during web navigation. Experimental results demonstrate that this mechanism substantially improves agent performance across multiple categories. Notably, the largest relative gains are observed in cross-paper comparison tasks, highlighting the importance of selective context retrieval for multi-document reasoning. Overall, our study provides a reproducible foundation for web-agent evaluation and offers insights into building more adaptive agents.

6 Limitation

One limitation of our benchmark is its exclusive focus on the English-language interface of the arXiv platform. This design choice overlooks multilingual versions of the site, which may present different navigation behaviors for non-English users. As a result, the benchmark may not fully capture the challenges faced by web agents operating in multilingual contexts. Expanding the benchmark to include tasks in other languages or region-specific interfaces would improve the generalizability of the benchmark.

7 Ethics Statement

This work introduces a benchmark for evaluating multimodal web agents on time-invariant tasks derived from the arXiv platform. All experiments were conducted on publicly available webpages without requiring user authentication or access to private data. No personal, sensitive, or user-generated information was collected or processed during the study. The benchmark tasks are carefully designed to avoid topics that could be ethically sensitive or controversial.

Human annotators involved in verifying task outcomes were fully informed of the study's goals and provided explicit consent. Annotations were limited to factual assessments of agent performance and did not require subjective judgments about individuals or user behavior.

References

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Lacoste, Massimo Caccia, Alexandre Drouin, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Graham Neubig, Quentin Cappart, Russ Salakhutdinov, and Nicolas Chapados. 2025. The browsergym ecosystem for web agent research. *Transactions on Machine Learning Research*. Expert Certification.

Google DeepMind. 2024. Gemini 1.5: Technical overview.

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. https://arxiv.org/abs/2306.06070. ArXiv:2306.06070.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11642–11662. PMLR.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2024. Multimodal web navigation with instruction-finetuned foundation models. In *International Conference on Learning Representations (ICLR)*.

Siddhant Garg, Harshita Bansal, Yihan Wang, Daniel Khashabi, and Ashish Sabharwal. 2025. Real: Benchmarking autonomous agents on deterministic simulations of real websites. *arXiv preprint arXiv:2504.11543*.

Petko Georgiev, Rohan Anil, and et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, and 1 others. 2025a. Mind2web 2: Evaluating agentic search with agent-as-a-judge. *arXiv preprint arXiv:2506.21506*.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025b. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*. Accepted to ICLR 2025 (Oral).

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. *arXiv preprint arXiv:2401.13919*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*. CVPR 2024 (Highlight).

Iat Long Iong, Xiao Liu, Yuxuan Chen, Hanyu Lai, Shuntian Yao, Pengbo Shen, Hao Yu, Yuxiao Dong, and Jie Tang. 2024a. Openwebagent: An open toolkit to enable web agents. In *ACL Demo Track*.

Iat Long Iong, Xiao Liu, Yuxuan Chen, Hanyu Lai, Shuntian Yao, Pengbo Shen, Hao Yu, Yuxiao Dong, and Jie Tang. 2024b. Openwebagent: An open toolkit to enable web agents on large language models. In *ACL 2024 System Demonstration Track*.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*. Accepted to ACL 2024.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2025. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*. Version 5, August 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen

Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*. Published in ICLR 2024.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *arXiv preprint arXiv:2401.13178*. NeurIPS 2024 (Oral).

Reiichiro Nakano and 1 others. 2023. Webgpt: Browser-assisted question-answering with human feedback. In *ICLR*.

OpenAI. 2024. Gpt-4o technical report. https://openai.com/index/gpt-4o. Accessed: May 2024.

Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024a. Autonomous evaluation and refinement of digital agents. *arXiv 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. *arXiv preprint arXiv:2406.12373*. Version 3, July 2024.

Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. 2023. From pixels to ui actions: Learning to follow instructions via graphical user interfaces. *arXiv preprint arXiv:2306.00245*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, 2 edition. MIT Press.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*. ACL 2023 camera ready, 23 pages, 9 figures, 11 tables.

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

Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, Ying Wen, Meng Fang, Muhao Chen, and 1 others. 2025. Agentic web: Weaving the next web with ai agents. *arXiv preprint arXiv:2507.21206*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*.

Danqing Zhang, Balaji Rama, Jingyi Ni, Shiying He, Fu Zhao, Kunyu Chen, Arnold Chen, and Junyu Cao. 2025a. Litewebagent: The open-source suite for vlm-based web-agent applications. *Preprint*, arXiv:2503.02950.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2025b. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23378–23386.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2023. Seeact: A multi-modal agent for web navigation with visual perception and action. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024a. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024b. GPT-4V(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Shuyan Zhou, Frank F Xu, Haozhe Li, Hang Lv, Amanpreet Singh, Alexander Ratner, Anca Dragan, and Chelsea Finn. 2024. Webarena: A realistic web environment for building autonomous agents. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. https://arxiv.org/abs/2307.13854. ArXiv:2307.13854.

A Appendix

This appendix provides additional details on task examples in Appendix A.1, pseudocode for the memory mechanisms in Appendix A.2, and agent prompts in Appendix A.3.

A.1 Task Example

Step 1: Enter Arxiv Website | Step 2: Search paper 2512.23227v1 | Step 3: Open Html version

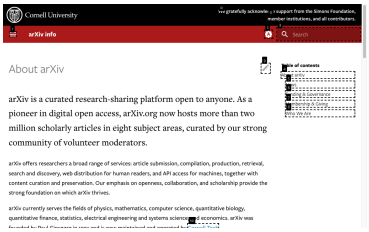
Step 4: Click to Experiment Section | Step 5: Scroll [WINDOW] | Step 6: Scroll/Count subsections

Step 7: Return to Google search | Step 8: Enter back to ArXiv | Step 9: Search paper 2512.22550v1

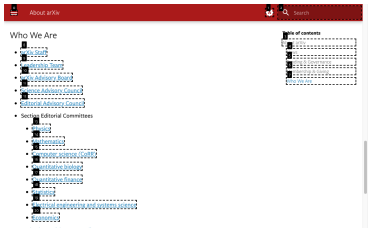
Step 10: Open Html version | Step 11: Click to Experiments Section | Step 12: Scroll [WINDOW]

Step 13: Scroll [WINDOW] | Step 14: Scroll [WINDOW] | Step 15: Compare counts/ANSWER

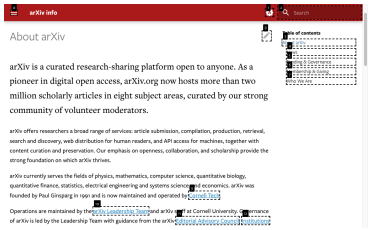
Figure 2: A cross-paper structural comparison task on arXiv. Given the task: "Which paper includes more Experiment subsections: 2512.23227v1 or 2512.22550v1?", the agent systematically navigates to both papers, inspects the Experiment sections, and compares their subsection structures. The agent correctly determines that both papers contain the same number of Experiment subsections, demonstrating accurate multi-document analysis and structural reasoning.



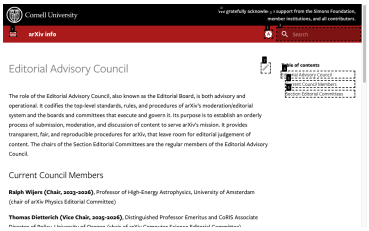
Step 1: Click [8]



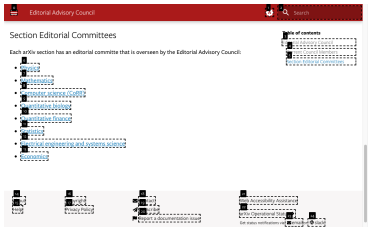
Step 2: Click [3]



Step 3: Click [11]

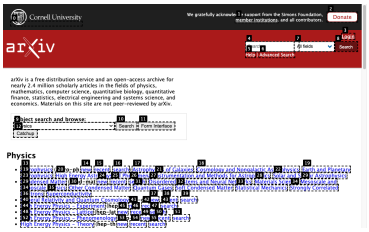


Step 4: Click [6]

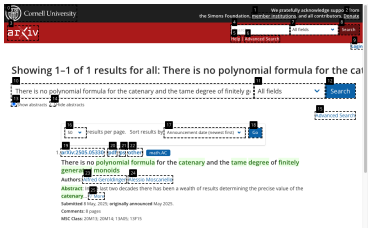


Step 5: ANSWER

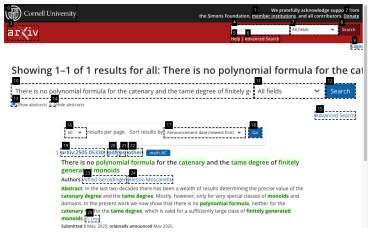
Figure 3: An organizational information retrieval case for arXiv. Given the task: “On arXiv’s About page, find the categories of the Section Editorial Committees.” The agent successfully retrieves the answer: “Physics, Mathematics, Computer science (CoRR), Quantitative biology, Quantitative finance, Statistics, Electrical engineering and systems science, Economics,” correctly identifying all eight top-level research domains designed by the platform’s editorial structure.



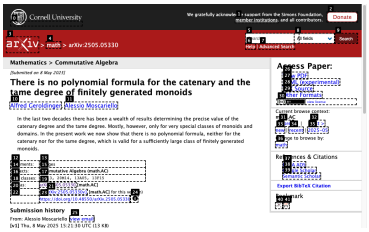
Step 1: Search paper



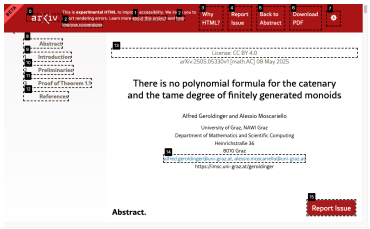
Step 2: Click [25]



Step 3: Click [19]

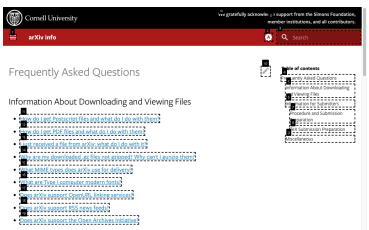


Step 4: Click [27]

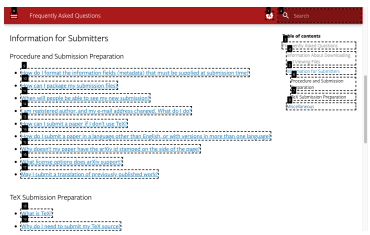


Step 5: ANSWER

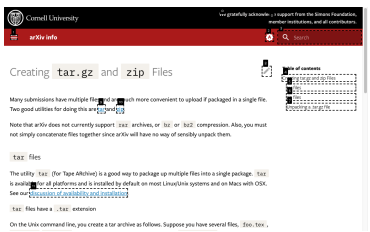
Figure 4: A publication detail retrieval task on arXiv. Given the task: “Provide the name of the university publishing in this paper: There is no polynomial formula for the catenary and the tame degree of finitely generated monoids.” The agent correctly extracts the affiliation information and returns: “University of Graz,” confirming successful deep content extraction from the publication metadata.



Step 1: Click [6]

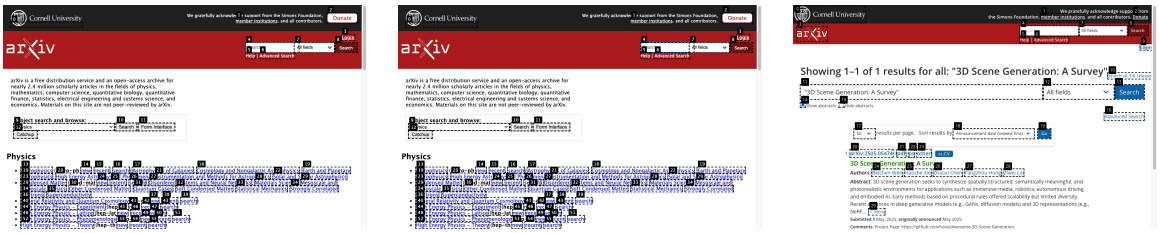


Step 2: Click [10]



Step 3: ANSWER

Figure 5: A user account management task on arXiv. Given the task: “How can I package my submission files?” The agent correctly returns the instruction: “Create tar.gz and zip Files,” accurately capturing the recommended submission packaging methods outlined in the official arXiv help documentation for authors preparing their papers.



Step 1: Click [4] Step 2: Type [4] Step 3: ANSWER

Figure 6: A paper discovery task on arXiv. Given the task: "State the name of the second Author of this paper: 3D Scene Generation: A Survey." The agent successfully identifies the second listed author as "Haozhe Xie," confirming the correct retrieval of metadata related to the specified research paper.

Step 1: Scroll [WINDOW] Step 2: Scroll [WINDOW] Step 3: ANSWER

Figure 7: A search interaction task on arXiv's advanced search page. Given the task: "Tell me how to search within a subcategory." The agent correctly interprets the search interface and returns the instruction: "Select All fields," demonstrating its ability to navigate and extract advanced search instructions from the user interface.

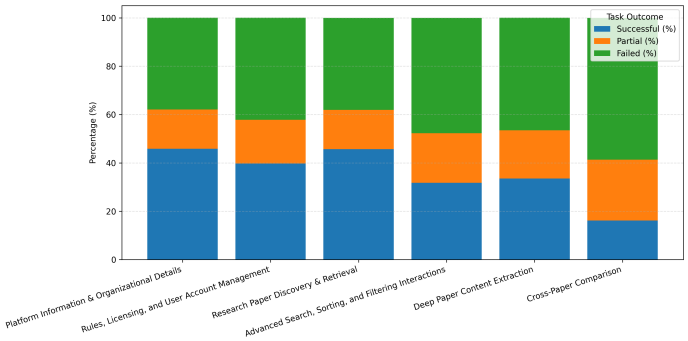


Figure 8: Stacked bar chart of GPT-o1, showing task completion rates across six task categories.

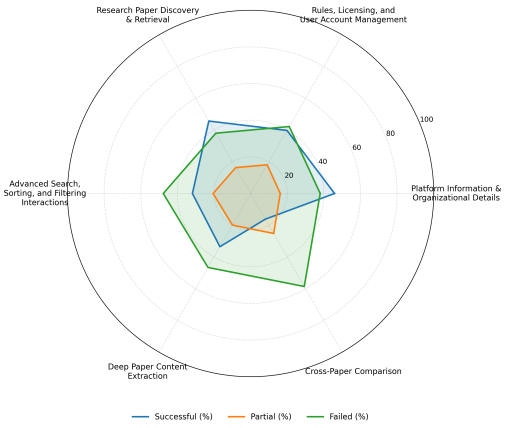


Figure 9: Radar chart of GPT-o1, visualizing success, partial, and failure rates across six task categories.

Web Agent	Success Rate (%)	Avg. Step Count	Trajectory Consistency ↑
GPT-o1	**54.1**	**8.6**	**0.82**
GPT-4-Turbo	36.4	9.4	0.71
GPT-4o	32.7	9.2	0.68
GPT-4o-mini	43.8	8.8	0.75
Gemini-1.5-pro	42.9	8.9	0.74
Gemini-2.0	30.6	9.7	0.66
Gemini-2.5	**53.7**	**8.9**	**0.78**
SeeAct	23.6	11.3	0.61
LiteWebAgent	44.0	9.0	0.74
OpenWebAgent	33.8	10.5	0.67

Table 5: Performance comparison of representative web agents across success rate, average step count, and trajectory consistency.

Model	Fixed History(%)	Random History(%)	Adaptive History(%)	Avg. Step Reduction↓
GPT-o1	43.2	41.0	52.4	-2.1
Gemini-2.5	47.3	45.2	**54.8**	-1.9

Table 6: We conducted an ablation analysis comparing three history retrieval strategies: (a) fixed-history (last 3 steps), (b) random-history (3 random past steps), and (c) adaptive-history (our dynamic reflection). Across all task categories, adaptive-history consistently outperforms both baselines. For example, GPT-o1 improves from $43.2\% \rightarrow 52.4\%$ ($+9.2\%$), while Gemini-2.5 rises from $47.3\% \rightarrow 54.8\%$ ($+7.5\%$). Efficiency also improves, with GPT-o1 requiring 2.1 fewer steps per successful task on average compared to fixed-history.

Step	Screenshot	Thought (with dynamic memory)	Action	Action target (button/text)
1	screenshot1.png	To find the university, I should search the paper by its title. (Initial step — no prior context to reflect on.)	Type [4]	Search bar [4]; input: "There is no polynomial formula for the catenary and the tame degree of finitely generated monoids"
2	screenshot2.png	Search results are shown. From Step 1, I confirmed this is the correct paper. Let's open more details.	Click [25]	More [25]; expands metadata and abstract
3	screenshot3.png	(Dynamic memory: Step 2 showed author names, but no affiliations.) To get structured metadata, we should access the article detail page.	Click [19]	arXiv:2505.05330 [19]; opens paper's detail page
4	screenshot4.png	(Dynamic memory: Step 3 showed abstract, still no affiliations.) The HTML version might include full author info and contact details.	Click [27]	HTML (experimental) [27]; opens the full paper in HTML format
5	screenshot5.png	(Dynamic memory: Step 4 gave full text. University found in both email domain and author line.)	ANSWER; University of Graz	From author info: alfred.geroldinger@uni-graz.at; full text shows: University of Graz

Table 7: Interaction trajectory for Task ArXiv–51 with dynamic memory. The agent uses targeted recall to improve navigation and stability across dense UI structures.

A.2 Pseudocode for Dynamic Memory Module

For each interaction step t:

1. Retrieve the last 3 visual observations and their associated element texts:
   `last_3_steps = get_last_3_steps()`
2. Ask the model which of these steps is most useful for reasoning:
   `memory_prompt = format_memory_prompt(last_3_steps)`
   `important_step_index = model.respond(memory_prompt)`
3. Construct the reasoning context:
   Reasoning source: `last_3_steps[important_step_index]`
   Current view: `last_3_steps[-1]`
4. Ask the model to generate the next action using both reasoning and current view:
   `action_prompt = format_action_prompt(task, guidance_context, history)`
   `next_action = model.respond(action_prompt)`
5. Execute the action and update history:
   `result = execute(next_action)`
   `update_history(next_action, result)`

Figure 10: Pseudocode for dynamic memory agent across the last 3 steps in WebArxiv.

A.3 Agent Prompt

SYSTEM_PROMPT = """
You are an intelligent web-browsing agent equipped with dynamic memory capability. Your task is to interact with webpages step-by-step to complete a given query efficiently and accurately.
At each step t, you will: 1. **Recall the last three interaction steps**, including screenshots and visible element texts. 2. **Reflect** on which past step is most relevant to current reasoning. 3. Use the **most relevant past step** as the reasoning context and the **latest observation** as the current view. 4. Generate the **next best action** that moves the task forward.

– Memory Phase – You will first receive up to three past observations (each includes a brief text summary and visible elements). Analyze these and identify **which step provides the most useful information** for solving the current task.
Response format: Memory: Step [1–3] is most relevant because [reason].

– Action Generation Phase – You will receive: 1. The reasoning source (from your selected past step) 2. The current view (most recent screenshot + element texts) 3. The task description and guidance context
Using these, decide the next best action.

– Allowed actions (choose ONE): – Click [Numerical_Label]: Click a labeled element. Type [Numerical_Label]; [Content]: Type text into a labeled textbox. Scroll [WINDOW]; [up/down]: Scroll the page. Wait: Wait if a page is still loading. GoBack: Navigate back to the previous page. ANSWER; [Your Answer]: Provide a final answer once the task is complete.

– Reasoning Principles – 1. Focus first on visible content before scrolling or navigating. 2. Use your chosen reasoning source to interpret the current page logically. 3. Avoid repeating the same action if no visual change occurred. 4. Use concise, goal-driven reasoning – no guessing or vague actions.

– Response Format – Your output must follow this structure: Memory: [Your analysis selecting the most relevant previous step] Thought: [Brief reasoning combining reasoning source and current view] Action: [Your chosen next action in the exact format above]
"""

Figure 11: System prompt for the dynamic memory web agent. The prompt encourages selective recall, contextual reasoning, and minimal action bias for structured web navigation.