MMInA: Benchmarking Multihop Multimodal Internet Agents

Shulin Tian[*] **Ziniu Zhang**[*] **Liangyu Chen**[*,†] **Ziwei Liu** ✉

S-Lab, Nanyang Technological University

{shulin002, ziwei.liu}@ntu.edu.sg

Abstract

Autonomous embodied agents live on an Internet of multimedia websites. Can they hop around multimodal websites to complete complex user tasks? Existing benchmarks fail to assess them in a realistic, evolving environment for their embodiment across websites. To answer this question, we present **MMInA**, a multihop and multimodal benchmark to evaluate the embodied agents for compositional Internet tasks, with several appealing properties: **1)** *Evolving real-world multimodal websites.* Our benchmark uniquely operates on evolving real-world websites, ensuring a high degree of realism and applicability to natural user tasks. Our data includes 1,050 human-written tasks covering various domains such as shopping and travel, with each task requiring the agent to extract multimodal information from web pages as observations autonomously; **2)** *Multihop web browsing.* Our dataset features naturally compositional tasks that require information from or actions on multiple websites to solve, to assess long-range reasoning capabilities on web tasks; **3)** *Holistic evaluation.* We propose a novel protocol for evaluating an agent's progress in completing multihop tasks. We experiment with both standalone (multimodal) language models and heuristic-based web agents. Extensive experiments demonstrate that while long-chain multihop web tasks are easy for humans, they remain challenging for state-of-the-art web agents. We identify that agents are more likely to fail on the early hops when solving tasks with more hops, which results in lower task success rates. To address this issue, we propose a simple memory augmentation approach that replays past action trajectories to reflect. Our method significantly improves the performance of both the single-hop and multihop web browsing abilities. Our code and data are available on github.com/shulin16/MMInA.

∗ Equal Contribution.   † Project Lead.   ✉ Corresponding Author.

1 Introduction

Building embodied agents capable of autonomous behaviors navigating in various environments has been a longstanding and intricate challenge in the realm of artificial intelligence research (Maes, 1993; Ziemke, 1998; Florian, 2003; Steels and Brooks, 2018). One common scenario that necessitates automation involves the interaction with digital interfaces (Puig et al., 2018; Toyama et al., 2021), with a particular emphasis on the automation of actions performed on rich Internet websites (Shi et al., 2017; Yao et al., 2023; Hong et al., 2024). Real-world web tasks are inherently compositional, requiring multihop actions across multiple websites. To accomplish this, agents must possess both *long-range planning* and *multimodal reasoning* capabilities. This includes understanding high-level instructions from user inputs, planning multihop actions across the web browser environment by leveraging HTML content and visual cues, and making informed predictions based on observations. While web agents perform well on single-hop tasks that require interactions with only one website, according to existing benchmarks (Li et al., 2023b; Liu et al., 2024, 2023b; Zhou et al., 2024; Koh et al., 2024), we observe that most agents struggle with multihop web tasks, which are prevalent in real-world scenarios where users must gather information from or take actions across multiple websites to accomplish a high-level task (Tab. 2). This gap motivates us to establish a multihop web browsing benchmark to assess the usefulness of Internet agents in natural multihop tasks.

Another gap in web agent research is multimodality. Existing benchmarks pose autonomous agent tasks that rely solely on textual information (Zhou et al., 2024; Deng et al., 2023; Yao et al., 2023). However, in real-world scenarios, visual information often plays an indispensable role and cannot be disregarded. For example, consider the
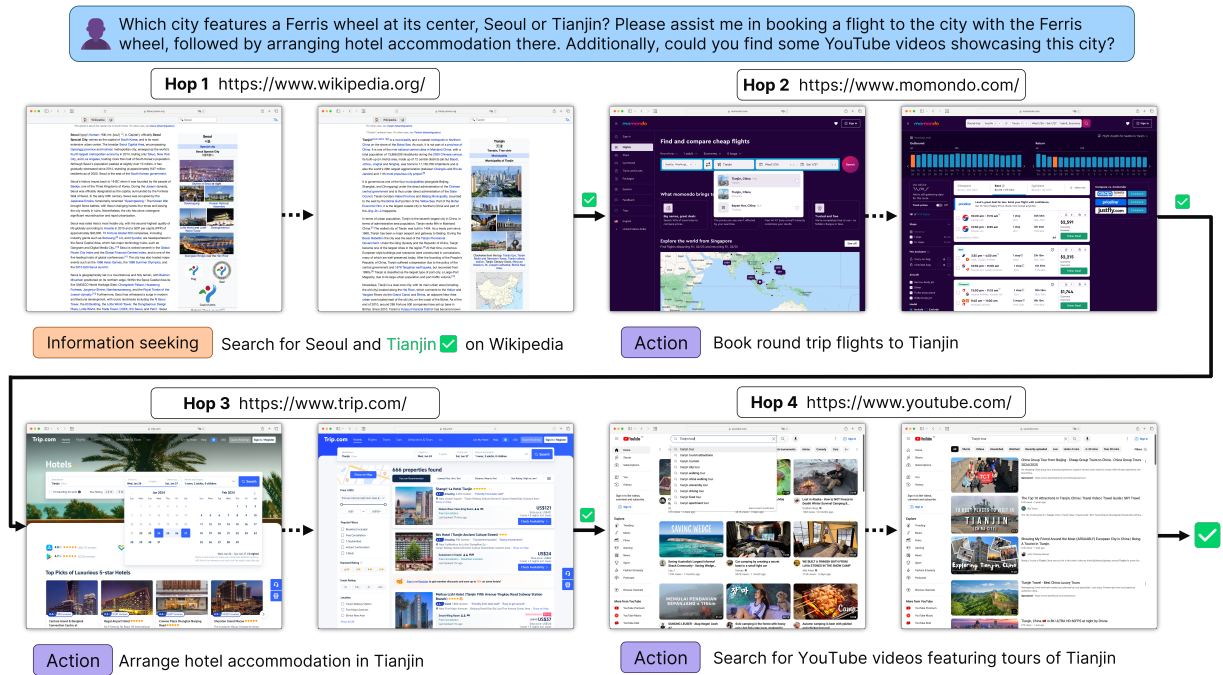
13682

Figure 1: **An example task from MMInA**. To evaluate an Internet agent's ability to carry out complex tasks, we make it navigate through a variety of websites to gather information and execute actions. In our proposed holistic evaluation protocol, each phase of the compositional task (defined as a *hop*) and the overall task are assessed for performance. Our benchmark includes 1,050 varied human-written multimodal tasks that require an average of 2.85 hops between websites and 12.9 actions to complete. The longest compositional task takes 10 hops.

task of "Help me purchase a blue cotton shirt", where the color attribute, derived from visual information, becomes crucial in fulfilling the user's request. However, there is a notable lack of current web-based agent benchmarks with emphasis on assessing the capabilities of comprehending and interacting with both textual and visual inputs, where a significant number of them primarily concentrate on tasks that involve text-based interactions.

To address the two issues above, we present MMInA, a novel benchmark designed to advance multihop and multimodal Internet task solving. MMInA operates on real-world, evolving websites (Fig. 1), offering high realism and applicability to natural scenarios (Tab. A3). It focuses on realistic tasks users commonly perform, such as navigating e-commerce platforms, extracting information from content-rich sites like Wikipedia, and conducting comparative analysis across web sources. Our 1,050 human-written tasks challenge agents to process multimodal inputs across multiple website hops and execute complex, multi-step reasoning, surpassing the simpler tasks found in existing benchmarks.

Our experiments with state-of-the-art models as agents' reasoning backbones show that while progress has been made in handling simple textual tasks, the integrated and sequential nature of tasks in MMInA remains a significant challenge. For

example, GPT-4V (Achiam et al., 2023) achieves a 21.8% success rate, which improves upon textual baselines but falls short of human performance (96.3%). Agents often fail early in multi-hop tasks (Tab. 3), leading to lower success rates. These results highlight the complexity of real-world web navigation and decision-making, underscoring the need for further advancements in multimodal and multihop reasoning. To bridge this gap, we propose a memory augmentation approach that replays past action trajectories, significantly improving both single-hop and multihop performance. This model-agnostic technique can be applied to other large multimodal models in the future.

In summary, our contributions are as follows:

- We introduce MMInA, an Internet agent benchmark featuring 1,050 multihop multimodal tasks across 14 diverse, evolving websites with realistic features. The multihop tasks assess more complex, human-like problem-solving actions, closely resembling real-world scenarios. We evaluate current large language models (LLMs) and large multimodal models (LMMs) as agents' backbones on the benchmark.

- We propose a holistic evaluation method for multihop tasks. Building on the limitations of task success rate evaluation, which often yields poor agent performance and limited

insights, we propose a new protocol for multihop tasks based on hop success rate. This provides a more granular assessment, offering deeper insights into the relationship between agent behavior and hop length, facilitating a more informed analysis of long-chain reasoning capabilities.

- We propose a lightweight and adaptable memory-augmented method that enhances agent performance by leveraging past action histories.

2 Related Works

Agent Benchmarks and Environments   Most existing works evaluate autonomous agents on curated textual I/O interfaces, leaving a gap in assessing their performance on real-world automation tasks. Tool-use benchmarks like (Li et al., 2023b; Patil et al., 2024; Liu et al., 2024) aim to assess agents' performances with tool-usage capability; BOLAA (Liu et al., 2023b) is another benchmark that coordinates multiple autonomous agents to make collective decisions. OpenAGI (Ge et al., 2023) and GAIA (Mialon et al., 2024) are multimodal benchmarks crafted for generalist agents that define multi-step tasks across multiple modalities. However, none of the above benchmarks explored the usage of LLMs or LMMs in web browsing environments or posed an effective evaluation metric specifically tailored for web agent tasks.

Web Agents   Webshop (Yao et al., 2022) builds a simulated e-commerce environment featuring 1.18 million real-world products, complemented by 12,087 crowdsourced textual instructions, while Mind2Web (Deng et al., 2023), CogAgent (Hong et al., 2024), and SeeAct (Zheng et al., 2024) try to construct a generalist web agent; WebVoyager (He et al., 2024) can automatically identify interactive elements based on the types of webpage. Recently, WebArena (Zhou et al., 2024) deploys a standalone set of multicategory websites in an interactive environment, where VisualWebArena (Koh et al., 2024) is a subsequent project that built upon WebArena, introducing the reliance on visual cues into the benchmark's design. However, we found that the tasks of existing benchmarks are oversimplified whose completions requiring a single website, which is highly diverged from the natural web browsing tasks and should originally be designed for multihop over a long-horizon setting.

3 MMInA Benchmark

3.1 Environment

Following Zhou et al. (2024), we formulate web browsing as a partially observable Markov decision process $\langle S, A, P, R \rangle$. The state space S is the whole Internet content, the status of the browser environment and agent, exceeding representable expressions in practice. Therefore, we pass the partial observation space Ω of S to the agent. At each time step t, an agent arrives at a certain state of a particular web page. The accessibility tree of the screenshot with linked images, with the action/state histories, forms a partial observation $o_t \in \Omega$ for the agent. Then the web agent takes an action a_t sampled from the action space A, being either an executable operation on the web page or a textual output as the answer to a question (Sec. 3.5). The state transition probability matrix $P : S \times A \rightarrow S'$ is implicitly encoded as the world knowledge of the Internet environment, which can be inferred or learned by a web agent. Our reward function R is expressed with language output, PASS or FAIL, as a result of each hop. Naturally, we define one hop as a subtask that is completed on a specific website. For example, in the task of Fig. 1, the agent receives a PASS if it finds the correct destination in the first hop, another PASS for arriving at the desired flight search page at the second hop, and so on.

Action Space   We follow Koh et al. (2024) to condense the potential agent-executed actions into a set of 12 summarized actions. Leveraging the playwright library, we simulate web pages on an X graphics server, employing a diverse array of actions to interact with the pages. These actions span a broad spectrum of behaviors mirroring human interactions with web pages, encompassing actions such as clicking on links, scrolling up and down using the scroll wheel, typing with the keyboard, etc. A higher hop count corresponds to a higher number of actions. On average, an MMInA task takes 12.9 actions to complete.

Observation Space   The observation space Ω usually embeds partial observations of the Internet to simulate real web browsing experiences. Observations include the task descriptions and the website content. To represent web content in a reasonable length, we primarily use accessibility trees that provide a structured and unified layout of the web page content. Each node of the accessibility

Benchmark	Multi-modal	Max / Avg. Hops	Website Type	Dynamic Interaction	# Websites
MiniWoB++ (Liu et al., 2018)	✓	1 / 1.00	Static simplified websites	✓ (Open-ended)	100
WebShop (Yao et al., 2022)	✓	1 / 1.00	Static simplified websites	✓ (Open-ended)	1
Mind2Web (Deng et al., 2023)	✗	1 / 1.00	Static real-world websites	✗ (MC)	131
RUSS (Xu et al., 2021)	✗	2 / 1.10	Static real-world websites	✗ (MC)	22
WebArena (Zhou et al., 2024)	✗	2 / 1.06	Static real-world websites	✓ (Open-ended)	6
VWA (Koh et al., 2024)	✓	2 / 1.05	Static real-world websites	✓ (Open-ended)	3
WebVoyager (He et al., 2024)	✓	4 / 2.40	Dynamic real-world websites	✓ (Open-ended)	15
MMInA	✓	10 / 2.85	Evolving real-world websites	✓ (Open-ended)	14

Table 1: **Comparison between the MMInA benchmark and related benchmarks.** MMInA employs a flexible environment that supports agents to generate open-ended actions. We selected 14 evolving real-world websites to benchmark multihop multimodal Internet agents, which can be easily expanded for future deployments.

a **Source websites of the 2,991 hops.**

b **Intent types of hops.**
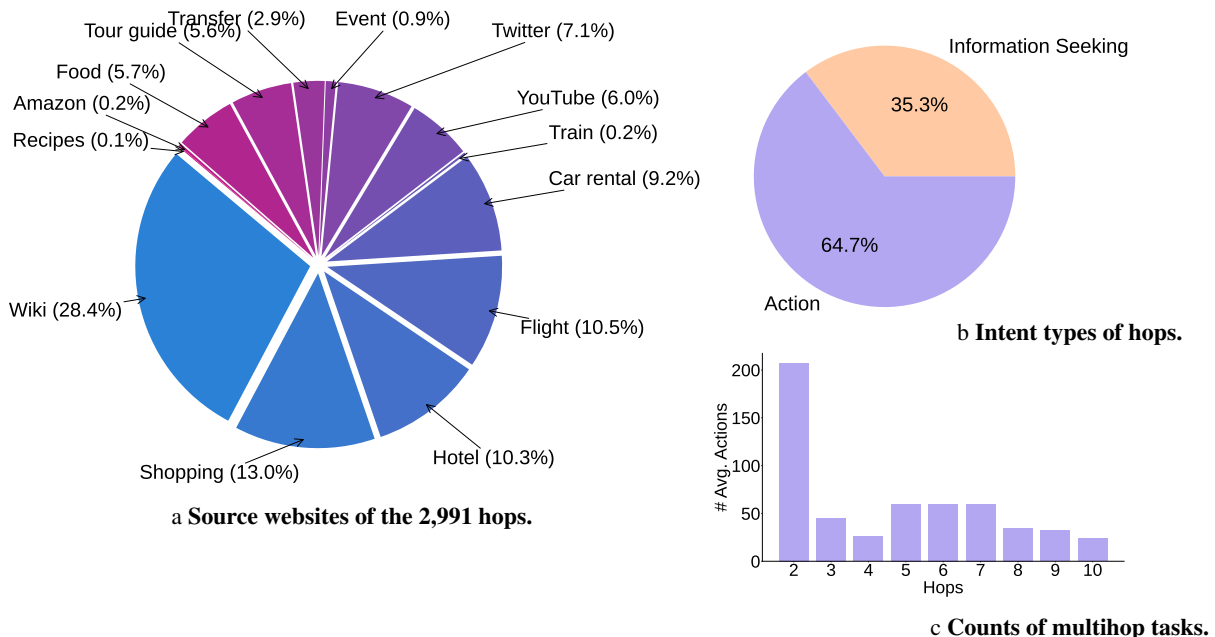
c **Counts of multihop tasks.**

Figure 2: **Statistics of the MMInA benchmark.** A web browsing task is composed of one or multiple hops between websites. MMInA covers diverse intents and domains of hops to resemble naturally compositional user tasks.

tree has an element ID, the type of the element, and the textual content of the element. If the element is an image, the environment downloads the image and paints the element ID on the image as a reference for the multimodal agent.

3.2 Dataset Construction

Data Structure MMInA adapted question styles from the WebQA dataset and used GPT-4V to generate similar questions with multimodal content (*e.g., "help me buy a yellow jacket online"*). The prompt structure includes: *1) Instructions:* some basic concepts about tasks, accessibility tree, and actions; *2) Rules:* such as "do nothing after action [stop]"; *3) QA examples:* some tiny examples are here to help the agent to understand instructions and above; *4) Reference Websites:* this is the universe of all potential websites the agent may visit; *5) Task:* a multihop multimodal task to solve. The hops vary from 1 to 10. Additionally, we manually crafted questions to diversify style, scope, and content across categories like shopping,

search, and booking. In summary, MMInA tasks combine 2,989 hops from 14 dynamic, real-world websites to ensure a diverse and realistic set of challenges.

Annotators & Annotation Protocols Annotators are human experts in web browsing and vary in age and gender to ensure the fairness of labeling. The annotators first proposed task templates varying in intent and difficulty. Each template generates $2 - 10$ tasks. All annotators then followed a "minimalist" approach, where annotators, acting as "omniscient readers", completed tasks using the shortest paths with all crucial website nodes recorded. This annotation protocol enhances the trajectory diversity in the evaluation process, where the ground-truth trajectory should be a subset of any successfully visited trajectories of the agents.

Performance Metrics Evaluating real websites is challenging due to their dynamic nature, as web content frequently changes. To address this, we propose a new evaluation metric for multihop tasks

based on the visited URLs. A task is considered successfully completed only when all required websites are visited in order, ensuring alignment between the agent's actions and the task objectives. Details are shown in Sec. 3.5.

3.3 Multimodal Web Content

Our work at MMInA focuses on multimodality-reliant tasks, which require both images and textual data to complete. For example, the task *"Which one is more furry, Hi&Yeah Comfy Faux Fur Cute Desk Chair or Armen Living Diamond Office Chair?"* requires the agent to locate and compare specified items on referenced web pages, analyzing the images and textual web page content to provide an answer. MMInA's approach contrasts with VWA, as all tasks in our framework necessitate the processing of both visual and textual information in multiple turns (Tab. A3).

MMInA includes an automated process for extracting accessibility trees from web pages while identifying and downloading the image contents in the current view. This allows agents to use images alongside the accessibility tree as inputs, highlighting the critical interaction between visual and textual data in solving real-world tasks within a multimodal framework.

3.4 Multihop Cross-website Browsing

MMInA dataset features multihop tasks across 14 distinct websites, covering diverse domains such as shopping, ticket booking, travel guides, and local food discovery (Fig. 2). A "multihop task" requires actions across multiple websites, with the agent automatically moving to the next site after completing each hop. This setup mirrors the complexity and real-world relevance of multihop web browsing, simulating the sequence of actions a human user would typically perform when tackling a high-level task. Task descriptions include links to the available websites (see Appendix B for details).

Figure 3: **Average actions in multi-hop tasks.**

3.5 Evaluation

Single-hop Evaluation Following Zhou et al. (2024), we implemented two evaluation method-

ologies for single-hop tasks to assess the semantics and effectiveness of predicted actions within the MMInA dataset. These methods offer either stringent or lenient criteria, depending on the task's characteristics.

The first method, "must_include", adopts a keyword-based evaluation approach. For each task, a set of essential keywords is defined. An agent's response is considered successful (PASS) if it incorporates all these keywords. Missing any keyword results in a failure (FAIL), ensuring a rigorous assessment based on keyword inclusion. The second method, "fuzzy_match", utilizes the capabilities of advanced language models, such as GPT-3.5-Turbo, to evaluate responses. It compares the agent's response and a reference answer by asking the model: "Given the statement pred, would it be correct to infer reference? Yes or No", where pred is the agent's response and reference is the reference answer. The evaluation is determined by the model's reply: a PASS is recorded if the answer is "Yes", and a FAIL if "No". This method offers a flexible evaluation framework that accommodates linguistic nuances, such as assessing semantic similarities between "gold" and "yellow" in color identification tasks.

Multihop Evaluation In our experiments with multihop tasks (Tab. 2), we often observe a remarkably low success rate for the entire task, if not zero. To provide a more granular evaluation, we propose an evaluation method tailored for N-hop problems: The evaluation involves maintaining a queue containing the conditions of each hop's completion. In particular, the last element of a queue is always an "END" marker that signifies the whole multihop task is completed, making the queue's length $N + 1$. An agent succeeds at a hop once it finds out the required information (e.g., an answer string) or reaches the desired state (e.g., a specific URL). For simplicity, our benchmark enforces that the agent completes tasks in sequence, i.e., the agent is only allowed to proceed to the next hop if the current hop is correctly completed. A task is completed only if all the hops are correctly completed in sequence.

Our interleaved single-hop and multihop evaluation methods aim to provide a systematic and insightful approach to assess the performance of agents in tackling multihop tasks, addressing the challenges posed by such tasks' complexity.
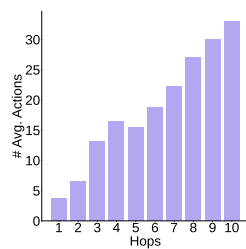
4 Experiments

4.1 Baselines

We employed a variety of state-of-the-art LLMs, LMMs, and adapted web-oriented models to evaluate their performance on the MMInA benchmark. For **text-based** models, we conducted evaluations in two settings: 1) *text-only*: Only textual information from the website was used, ignoring image content. 2) *caption-augmented*: In addition to the text, we used the BLIP-2 (Li et al., 2023a) model to generate captions for website images, incorporating visual information. For **multimodal** models, both image and text information from the website were provided. More details regarding the environment, model parameters, and versions are included in Sec. A. We categorized the models as follows:

1) LLM/LMM as Agent's Reasoning Backbone
Large language / multimodal models can act as powerful backbones in agents' reasoning processes that can predict feasible next-step actions by prompting (Liu et al., 2023b; Zhou et al., 2024). As the textual input is the accessibility tree representation of the webpage, we categorized the text-based agents into 4 groups: 1) pretrained open-sourced LLMs, like CodeLLaMA(Roziere et al., 2023) and DeepSeek-R1 (Guo et al., 2025); 2) text decoders from pretrained open-sourced LMMs, like Fuyu-8b (Bavishi et al., 2023); 3) API-based LLMs, like GPT-4 (Achiam et al., 2023) and Gemini-Pro (Team et al., 2023). Our experiments also involved prominent LMMs such as Fuyu-8b (Bavishi et al., 2023), Gemini-Pro-Vision (Team et al., 2023), and GPT-4V (Achiam et al., 2023).

2) Heuristic-Based Web Agents Several heuristic-based web agents were specifically crafted with the intention of navigation and completion of web-based tasks (Yao et al., 2022; Deng et al., 2023; Hong et al., 2024; Zheng et al., 2024). We selected WebShop and CogAgent as baselines to evaluate how models trained on web-based tasks perform on the MMInA benchmark.

3) Human Baselines We conducted a comparison of hop and task performances within the same settings with an average of 3 human test takers. The test takers come from various socioeconomic backgrounds, without information on the tasks before the evaluation. Human baselines consistently outperform all existing web agents with significant margins.

4.2 Main Results

The results for the different models are shown in Tab. 2, where the hop performance and task performance are evaluated respectively. The hop success rate reflects the percentage of successful visits to targeted websites, while the task success rate measures the percentage of tasks successfully completed by agents out of the total number of tasks.

Our experimental results indicate that current state-of-the-art models exhibit significantly reduced performance on multihop tasks. This gap highlights their struggle to effectively recognize and process structured, long-context information from web pages, which is essential for understanding web content. Additionally, performance drops sharply as the number of hops increases, revealing the models' limitations in long-chain reasoning.

The hop success rate, which counts every successful task completion at a website, serves as an auxiliary metric to represent the procedural performance of each agent more accurately. On single-hop tasks, DeepSeek-R1-Distill-Qwen-32B outperformed all other models, indicating that the reasoning model possesses strong potential in image and context comprehension, as well as planning. However, for tasks with a hop count ranging from 2 to 4, we observed unexpected performances. Specifically, Gemini-Pro without captions and GPT-4 without captions exhibited higher performances compared to their counterparts that are augmented with captions. Further analysis of the agents' trajectories revealed that when agents were assigned relatively simple tasks while being under-informed or lacking sufficient information, they tended to "wander" through the given hops. This often led to an endless loop, causing them to lose focus on the task's original intent and ultimately resulting in failure. This phenomenon explains why some agents achieve a higher hop success rate while simultaneously exhibiting a low task success rate. This insight further justifies the need for a holistic evaluation protocol with MMInA.

The experimental results showed that: 1) *Multimodality-reliance*: Multimodal models exhibit overall higher performance in both hop and task performance, which makes more accurate predictions on the proposed benchmark; 2) *Context window length*: Language models like CodeLLaMA and the GPT series, designed for structured and long-context processing, excel in web-based tasks relying on structured webpage representa-

Input Type	Agent	Inputs	Hop Success Rate (↑)				Task Success Rate (↑)			
			1 hop	2-4 hops	5+ hops	overall	1 hop	2-4 hops	5+ hops	overall
Text	Fuyu-8B		0	0	0	0	0	0	0	0
	CodeLLaMA-7B		1.18	0	0	0.29	1.18	0	0	0.58
	WebShop	📄	20.67	0	0	4.17	20.67	0	0	10.12
	DeepSeek-R1-Distill-Qwen-32B	</>	21.61	1.85	1.62	4.74	21.61	0	0	10.46
	Gemini-Pro		19.09	**34.12**	2.13	11.85	19.09	0.76	0	9.54
	GPT-4		14.37	30.56	5.23	12.26	14.37	9.09	0	9.34
	CodeLLaMA-7B		5.71	0	0	1.61	5.71	0	0	2.79
	WebShop	📄	29.72	0.00	0.00	5.61	29.72	0	0	14.55
	DeepSeek-R1-Distill-Qwen-32B	</>	**47.68**	3.84	4.68	11.11	**47.68**	0	0	23.07
	Gemini-Pro	CC	30.12	11.09	0.05	12.38	30.12	1.52	0.38	15.22
	GPT-4		38.58	20.70	3.43	13.50	38.58	3.79	0	19.85
Multimodal	CogAgent-9B		6.92	0	0	1.06	6.92	0	0	3.35
	GPT-4o	📄	21.90	9.23	0.96	5.94	21.90	3.85	0	11.61
	Fuyu-8B	</>	27.36	0	0	5.52	27.36	0	0	13.39
	Gemini-Pro-Vision	🖼	28.94	16.38	4.03	10.66	28.94	1.51	**1.13**	18.40
	GPT-4V		42.91	21.23	3.99	13.89	42.91	3.03	0	**21.77**
	GPT-4o	📄 </>	27.45	17.76	**10.13**	**14.36**	27.45	3.32	0	14.04
	Gemini-Pro-Vision	🖼 ⏪	39.17	23.93	4.78	14.27	39.17	**10.61**	**1.13**	20.13
-	Human	🖥	99.02	97.91	93.77	98.43	99.02	95.34	88.12	96.25

Table 2: **MMInA Benchmark Results.** We evaluated 4 types of agents on the proposed MMInA benchmark: 1) LLM Agents; 2) LMM Agents; 3) Heuristic-Based Web Agents; 4) Human Baselines. Regarding to different capabilities of agents, we have different combinations of input types. Here are the definitions of all types of input: 📄: input instructions; </>: the accessibility tree of the current webpage; CC: the textual captions of the images in the current view; 🖼: the images in the current view; ⏪: the execution histories of the agent; 🖥: the original webpage. The hop success rate is defined by the percentage (%) of successful visits to the targeted websites; while the task success rate is calculated by the overall percentage (%) of successful tasks from the whole task set.

tions. Reasoning models like DeepSeek-R1 perform well in single-hop tasks due to strong contextual comprehension. However, when tackling multi-hop tasks that require retaining longer contexts, R1 struggles and exhibits degraded performance; 3) *Web-based models*: the models that were trained on web-based content (e.g., Fuyu-8B, WebShop) still exhibit the versatility and adaptability in unfamiliar environments.

4.3 Analysis: Why are Multihop Web Tasks Challenging?

Search Space   Agents often underperform on multihop tasks, failing at early hops, yet excel when each hop is treated as an individual single-hop task. Our analysis shows that in single-hop tasks with a single reference URL, agents tend to repeatedly attempt actions within the same website upon failure until the task is completed. Conversely, in multihop tasks where prompts include multiple websites, agents that fail on the expected site often switch to alternate sites instead of persistently retrying. This behavior leads to excessive exploration and a significant drop in task completion rates.

For example, in a complex task like "Book a flight to Tokyo, find a tour guide, watch YouTube videos, rent a car, and book a hotel", agents may fail in the first step. Even if they correctly redirect to a relevant site, limited memory prevents them from recalling previous steps, leading to repeated actions without progress. Moreover, despite defined termination conditions for each hop, agents often fail to recognize and apply them, lingering in completed hops instead of advancing, ultimately failing the task.

Agent Input Length   The total hop count in a multihop task determines its length. Each hop's success depends on the previous one, but the success rate of any individual hop should be independent of the total number of hops, provided it remains within the specific domain. However, after aligning the first hop semantically across all tasks, our empirical findings in Tab. 3 indicated an unexpected pattern that was contrary to the assumption above. We observed that agents performed better on tasks with fewer total hops, achieving higher success rates in completing the first hop. Conversely, as the total hop count increased, there was a noticeable decline in the success rate within the first hop. We attribute this phenomenon to the enlarged search space and the agent's weak zero-shot long-context reasoning capabilities, which we resolve in Sec. 4.4.

These findings highlight the complexity of multihop tasks, which is not simply the sum of single-

SR(%) H.C.	1st	2nd	3rd	4th	5th	6th
2	56.50	11.00	-	-	-	-
3	22.73	4.55	0.00	-	-	-
4	12.50	0.00	0.00	0.00	-	-
5	12.28	1.75	0.00	0.00	0.00	-
6	16.67	0.00	0.00	0.00	0.00	0.00

a GPT-4V

SR(%) H.C.	1st	2nd	3rd	4th	5th	6th
2	69.28	8.43	-	-	-	-
3	32.56	0.00	0.00	-	-	-
4	40.00	0.00	0.00	0.00	-	-
5	41.67	5.00	0.00	0.00	0.00	-
6	31.03	1.72	0.00	0.00	0.00	0.00

b Gemini-Pro-Vision

Table 3: **Evaluation by Hops.** Tables (a) and (b) display the hop success rates (SR), distinguished by hop counts (H.C.) of the tasks ranging from 2 to 6, for the models GPT-4V and Gemini-Pro-Vision respectively. Higher success rates are marked with darker colors. Both agents fail on the early hops when solving tasks of more hops.
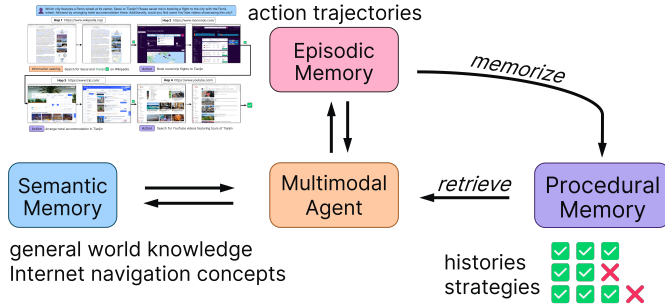

Figure 4: **Memory-augmented agents.** Our method complements LMMs by enhancing procedural memory with action trajectories on similar tasks.

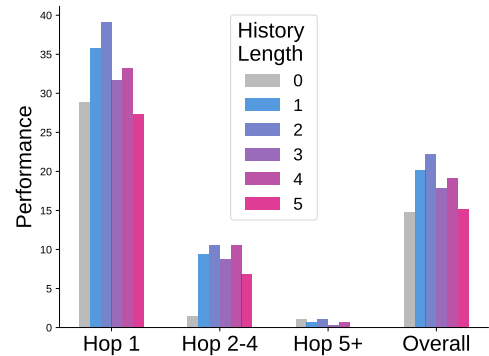Figure 5: **Success rates of memory-augmented agents, by history lengths.**

hop performances but involves intricate task flow management. This complexity underlines the necessity for web agents to possess advanced planning and reasoning skills to effectively navigate and execute multihop tasks.

4.4 Memory-augmented Agents

Agents in dynamic environments must make decisions based on real-time observations, user queries, and past trajectories. Our experiments reveal the complexity of action prediction, requiring different memory types at various stages. This highlights the need to retain information across tasks, actions, and web interactions. We propose memory-augmented web agents with three key memory systems: semantic, episodic, and procedural.

- Semantic memory stores the agent's general world knowledge, continuously updated from the Internet or knowledge bases, typically encoded in the weights of large language models.

- Episodic memory temporarily holds step-by-step action trajectories, enabling the agent to recall previous actions for ongoing tasks, often represented as context in autoregressive models or in-context examples.

- Procedural memory activates after completing a task, encoding the full action sequence and outcomes to refine strategies for future tasks.

Our results emphasize the role of procedural memory in improving agent performance by replaying past action trajectories for similar tasks (see Table 2). With these memory systems, multimodal agents can access and apply relevant information, enabling more sophisticated, contextually aware responses to dynamic environments, thereby significantly enhancing performance and adaptability.

5 Conclusion, Challenges, and Outlook

We present MMInA, a benchmark with three key features: **1)** It benchmarks agents on real-world websites with 1,050 multimodal multihop tasks across 14 diverse websites, including experiments with state-of-the-art LLMs and LMMs, as well as human baselines; **2)** It introduces a novel holistic evaluation method for multihop tasks, assessing both task-level and hop-level success rates; **3)** It proposes a flexible memory-augmented approach to enhance agents' performance by improving their procedural memory.

Future Work Moving forward, we will consider employing an evaluation method focused on

actions, which will directly guide the agent's operations.

6   Limitations

Due to the protection mechanisms employed by web pages, it's exceptionally challenging to find a website that allows us to directly fetch images from HTML files. So, one of the websites we utilized is an offline standalone website, and the other is an open-source website.

7   Ethical Considerations

Bias in the base multimodal models can lead to inaccurate or unfair outcomes. Users should consider the representativeness of training data to avoid biased behavior.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Introducing our multimodal models.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Biplab Deka, Zifeng Huang, and Ranjitha Kumar. 2016. Erica: Interaction mining mobile apps. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 767–776.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Razvan V Florian. 2003. Autonomous artificial intelligent agents. *Center for Cognitive and Neural Studies (Coneural), Cluj-Napoca, Romania*.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. 2023. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112.

Tianyu Gao, Zirui Wang, Adithya Bhaskar, and Danqi Chen. 2024. Improving language understanding from screenshots. *arXiv preprint arXiv:2402.14073*.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. 2023. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36:5539–5568.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. 2024. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*, pages 161–178. Springer.

Jihyung Kil, Chan Hee Song, Boyuan Zheng, Xiang Deng, Yu Su, and Wei-Lun Chao. 2024. Dual-view visual contextualization for web navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14445–14454.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel

Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Joshua Adrian Cahyono, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2025. Otter: A multi-modal model with in-context instruction tuning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023b. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3116.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations (ICLR)*.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, et al. 2023b. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *arXiv preprint arXiv:2308.05960*.

Xing Han Lu, Zdeněk Kasner, and Siva Reddy. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. In *International Conference on Machine Learning*, pages 33007–33056. PMLR.

Pattie Maes. 1993. Modeling adaptive autonomous agents. *Artificial life*, 1(1_2):135–162.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. Gaia: A benchmark for general ai assistants. In *International Conference on Learning Representations (ICLR)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Learning Representations (ICLR)*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *International Conference on Learning Representations (ICLR)*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pages 3135–3144. PMLR.

Luc Steels and Rodney Brooks. 2018. *The artificial life route to artificial intelligence: Building embodied, situated agents*. Routledge.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. 2021. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. In *Workshop of International Conference on Learning Representations (ICLR Workshop)*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, pages 54590–54613. PMLR.

Nancy Xu, Michael Du, Giovanni Campagna, Larry Heck, James Landay, Monica S Lam, et al. 2021. Grounding open-domain instructions to automate web support tasks. In *2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. 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. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Zijia Zhao, Longteng Guo, Tongtian Yue, Sihan Chen, Shuai Shao, Xinxin Zhu, Zehuan Yuan, and Jing Liu. 2023. Chatbridge: Bridging modalities with large language model as a language catalyst. *arXiv preprint arXiv:2305.16103*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning*, pages 61349–61385. PMLR.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*.

Tom Ziemke. 1998. Adaptive behavior in autonomous agents. *Presence*, 7(6):564–587.

A Experiment Details

A.1 Environment & Parameter Settings

We followed the display settings from (Zhou et al., 2024), using a viewport of 1280×2048, and provided the webpage accessibility tree as text input to the models. Default parameters were used from either open-source pre-trained models or API-based models. For the web-trained agent WebShop, which was designed for a static environment and trained with specially structured queries, we used the GPT-3.5-turbo model to generate formatted queries in place of those typically sourced from the built-in environment. For the versions of API-based models, the GPT-4 model in the paper is referred to `gpt-4-0125-preview`; the GPT-4o model is referred to `gpt-4o-2024-11-20`; the GPT-4V model is referred to `gpt-4-vision-preview`; the Gemini-Pro model is referred to `gemini-1.0-pro-001`; the Gemini-Pro-Vision is referred to `gemini-1.0-pro-vision-001`.

A.2 Computing Resources

We used 1 Nvidia RTX6000 Ada GPU with 48 GB memory for the pre-trained baseline models (most of them have 7b/8b/9b parameters) that run the inference code locally. The inference of each epoch on MMInA takes 4-8 hours, varying by model inference performance and forward times.

A.3 Supplementary Results

Hop Analysis   We follow the previous settings of hop analysis in the main paper, illustrating the agent performance of a GPT-4V agent in Tab. A1. We observed again that agents performed better on tasks with fewer total hops, achieving higher success rates in completing the first hop. Conversely, as the total hop count increased, there was a noticeable decline in the success rate within the first hop. Because there are fewer long-range (>7 hops) tasks, the success rates fluctuate due to randomness.

Ablation of agents' memory   We enhance LMMs with memory by appending action trajectories from the last K tasks, comprising task descriptions and web content observations, to their prompts. This approach, which integrates replayed experiences, helps narrow the agent's search space, thereby grounding its reasoning. However, this technique multiplies the input length by K, presenting a challenge for LMMs accustomed to shorter inputs. To find a balance, we determined the optimal K value for constructing procedural memory, which, as illustrated in is typically $K = 2$. Our tests show that agents with procedural memory enhancements perform better in action prediction and execution. Yet, we observed a non-linear relationship between the number of historical references and performance. In simpler tasks within domains like shopping or Wikipedia, a smaller historical set—specifically $K = 1$ or 2—tended to yield better results, while larger histories offered diminishing returns and introduced biases and disturbances into the decision-making process. Although these experiments were conducted using Gemini-Pro-Vision, our method is designed to be model-agnostic, adaptable to any LMM or LLM.

B MMInA Benchmark Details

B.1 Datasets

B.1.1 Annotators

MMInA is constructed by three human annotators from scratch. They, varying in age and gender but proficient in web browsing, were provided pre-annotated examples and followed consistent guidelines. Each labeled different dataset portions, and cross-validation was conducted for task diversity and answer accuracy. All annotators signed formal agreements and were trained for annotation.

B.1.2 Anontation Protocals

The final trajectory includes all necessary website nodes. Agents, however, can freely explore and visit unnecessary websites before completing tasks.

B.1.3 Data Statistics

From the hop counts shown in Fig. 3, we observe that as the number of hops increases, the number of actions required by the agent also increases. However, it's worth noting that the average number of actions required for 5-hop data is lower than that for 4-hop data. In our dataset, 4-hop content involves comparative operations, such as *"Which one has a Ferris wheel in the center of the city, Tianjin or Chengdu?"* Since our definition of multi-hop tasks involves navigation across different web pages, we do not categorize these comparative questions as 5-hop tasks.

Most of our multitasking revolves around travel. First, we need to determine the travel destination. We let the agent determine the travel destination by retrieving and answering questions on Wikipedia.

GPT4V	count	sr1	sr2	sr3	sr4	sr5	sr6	sr7	sr8	sr9	sr10
2-h	200	56.50	11.00								
3-h	44	22.73	4.55	0.00							
4-h	16	12.50	0.00	0.00	0.00						
5-h	57	12.28	1.75	0.00	0.00	0.00					
6-h	60	16.67	0.00	0.00	0.00	0.00	0.00				
7-h	59	25.42	0.00	0.00	0.00	0.00	0.00	0.00			
8-h	35	40.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
9-h	30	56.67	20.00	3.33	0.00	0.00	0.00	0.00	0.00	0.00	
10-h	19	52.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A1: **Performance of GPT-4V on multihop tasks based on the hop number ranging from 2 to 10.** The success rate (sr) is calculated based on single-hop evaluation results over the whole completed number of hops.

A set of tasks is related to travel, including booking flights, reserving hotels, exchanging currencies, etc. Then we randomly select some tasks from the task set and combine them into a complete and smooth task. In each task's JSON file, we include keywords such as "flight", which is the three-letter code of the destination airport. The reason for these keywords is that for specific tasks, some sub-tasks cannot be measured with a unified standard, so we add these keywords to judge the endpoints of specific tasks. Another type of task related to cooking includes purchasing food on Amazon and searching for recipes. Each task is close to real life and uses real and dynamic web pages for operation. We have 1050 tasks in total and each task contains a QA pair and other supportive materials. 108 QA pairs are filtered from WebQA (Chang et al., 2022). Statistical details about our dataset are shown in Fig. 2.

B.2 Tasks

B.2.1 Website Links

Tab. A2 reveals the links to each website defined in MMInA.

B.2.2 Comparision with other benchmarks

The comparison of our benchmark and the VWA benchmark is shown in A3.

C More Related Works

The literature review will be divided into several key sections, each focusing on a critical aspect of research related to the development of multimodal autonomous web agents. This comprehensive overview will delve into multimodal datasets, large language/multimodal models as backbones, and various types of autonomous agents, including embodied agents and web agents. The aim is to provide a thorough understanding of the current state of research in these areas, as well as to identify gaps and provide insights for future work.

Multimodal Datasets Recent progress in multimodal learning, showcased by models like CLIP(Radford et al., 2021), DALL-E(Ramesh et al., 2021), Stable Diffusion(Rombach et al., 2022), Flamingo(Alayrac et al., 2022), and GPT series, has led to significant improvements in areas such as zero-shot classification, image creation, and in-context learning. Although these models employ various algorithmic approaches, including contrastive learning, diffusion techniques, and auto-regressive models, they share a fundamental reliance on large datasets comprising image-text pairs. This commonality underscores the importance of such datasets in driving advancements in multimodal AI capabilities.

Webdataset[*] is a commonly used dataset as it contains thousands of image-text pairs data scraped from websites; LAION-5B(Schuhmann et al., 2022) is a dataset that contains 5.85 billion CLIP-filtered image-text pairs, where 2.32B contain English language; MIMIC-IT(Li et al., 2025) is a dataset consists of 2.8 million multimodal instruction-response pairs equipped with rich in-context information, with 2.2 million unique instructions derived from images and videos; DataComp(Gadre et al., 2023) is a newly brought dataset which consists of four stages: A) Deciding on a scale that fits within resource limitations. B) Creating a dataset, opting for either the filtering

---

[*]https://github.com/webdataset/webdataset

Table A2: **Links to MMInA websites.** All of them are evolving real-world websites with content updated over time, except Shopping. Our flexible evaluation protocol facilitates supporting more websites in the future.

Description	URL
Wikipedia[1]	`https://library.kiwix.org/viewer#wikipedia_en_all_maxi_2024-01/A/User%3AThe_other_Kiwix_guy/Landing`
Car renting	`https://www.trip.com/carhire/`
Flight booking	`https://www.momondo.com/`
Hotel booking	`https://www.trip.com/hotels/`
Event searching	`https://www.eventbrite.com/`
Twitter	`https://twitter.com/home`
Amazon	`https://www.amazon.com/`
YouTube	`https://www.youtube.com/`
Find food	`https://www.timeout.com/`
Exchange dollars	`https://www.xe.com/`
Travel guide	`https://www.nomadicmatt.com`
Recipes	`https://www.allrecipes.com/`
Train booking	`https://www.trip.com/trains/`
Shopping	OneStopMarket (an offline standalone website)

[1] Since libraries in Kiwix may update, resulting in URLs with advanced dates, it's advisable to verify the Wikipedia library on the official Kiwix page. However, this doesn't affect our experiments.
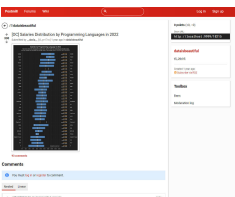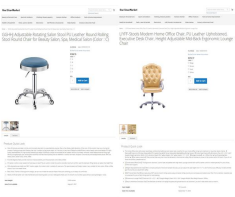
Websites	Task
VWA	
	When did the programming language that has the largest variance in salary first appear? Answer using the information from the Wikipedia site in the second tab.
MMInA	
	Do both LIYFF-Stools Modern Home Office Chair and GGHHJ Adjustable Rotating Salon Stool PU Leather Round Rolling Stool Round Chair have armrests?

Table A3: **Comparison between VWA (Koh et al., 2024) and MMInA.** MMInA requires multimodal inputs at multiple steps to accomplish the task, which makes it a more challenging multimodal benchmark.

approach or the Bring Your Own Data (BYOD) track. C) Using a set architecture and specific hyperparameters to train a CLIP model on the created dataset. D) Assessing the performance of the trained model across a variety of downstream tasks.

Large Language/Multimodal Models as Backbones   Instruction tuning is a common method used in LLM training, which involves refining pretrained LLMs using datasets formatted as instructions. This approach enhances the model's ability to perform new, unseen tasks by simply following directions, thereby improving its zero-shot capabilities. Some notable models like ChatGPT(Achiam et al., 2023), InstructGPT(Ouyang et al., 2022), FLAN(Wei et al., 2021; Chung et al., 2024) are built on top of instruction tuning methods.

Inherited the success from LLMs, LMM training is also extended to the instruction-tuning methods by utilizing the multimodal instruction data, which contains: a textual `<instruction>` to describe the task; `<image>`, `<text>` pair as input to enable the multimodalities; the model output with a token `<output><EOS>` to identify the end of the output. A multimodal instruction sample can be denoted in a triplet form, i.e., I, M, R, where I, M, R represent the instruction, the multimodal input, and the ground truth response, respectively. The LMM predicts an answer given the instruction and the multimodal input:

$$A = f(I, M; \theta)$$

, and the optimizing objective can be formulated

as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(R_i | I, R_{<i}; \theta) \qquad (1)$$

1) Transformation The efficacy of instruction tuning in the training of LMMs is significantly constrained by the limitations in length and type of data available in current Visual Question Answering (VQA) datasets. To address this, some researchers have opted to adapt the provided instructions, transforming the succinct answer data into extended sentences enriched with semantic details (Zhao et al., 2023). Other studies, such as in, reconstructed the answer by prompting ChatGPT to emulate the capabilities of advanced language models.

2) Self-Instruct LLaVA (Liu et al., 2023a) extends the multimodal approach by converting images into descriptive texts and outlines of bounding boxes, then uses GPT-4 to create additional data within the context provided by initial examples.

Autonomous Agents in Virtual World Agents designed for Graphical User Interfaces (GUIs) are crafted to streamline complex activities on digital devices like smartphones and desktops. These GUI agents may employ HTML as inputs or use screenshots to facilitate task execution in a broader context. Traditionally, research has revolved around training these agents in restrictive, static environments, a practice that deviates from human learning and hinders the agents' ability to make decisions akin to humans. However, the emergence of large language models (LLMs) and large multimodal models (LMMs) equipped with vast web knowledge marks a pivotal shift towards achieving a more human-like intellect in agents, sparking a surge in research on LLM/LMM-enhanced autonomous agents. This section aims to explore the latest state-of-the-art (SOTA) developments in autonomous agents, examining both web GUI agents and mobile GUI agents.

1) GUI Agents - Web Agents Despite the current progress of web agents discussed in the main paper, several works also explored the development of web agents. TravelPlanner (Xie et al., 2024) proposes a benchmark that provides a sandbox environment with tools for accessing nearly four million data records. It includes 1,225 planning intents and reference plans to evaluate the planning strategies of language agents by using tools; Omni-ACT (Kapoor et al., 2024) presents a dataset and

benchmark for assessing an agent's capability to generate executable programs for computer tasks. It uses the PyAutoGUI Python library to automate mouse and keyboard operations across different operating systems and web domains. It addresses the limitations of HTML-based agents by providing a multimodal challenge where visual cues are crucial, thus enabling a more robust understanding of UI elements, but it still shows the inability to handle native desktop applications or multi-application tasks; WEBLINX (Lu et al., 2024) also proposes a benchmark for conversational web navigation, addressing the problem of enabling a digital agent to control a web browser and follow user instructions in a multi-turn dialogue fashion. The method involves a retrieval-inspired model that prunes HTML pages by ranking relevant elements, addressing the issue of LLMs not being able to process entire web pages in real-time. The technology used includes a dense markup ranker for element selection and multimodal models that combine screenshots, action history, and textual website representation. The performance is evaluated on tasks like creating a task on Google Calendar, with the model's ability to replicate human behavior when navigating the web; DUAL-VCR (Kil et al., 2024) leverages the "dual view" of HTML elements in webpage screenshots, contextualizing each element with its visual neighbors. This approach uses both textual and visual features to create more informative representations for decision-making.

2) GUI Agents - Mobile Agents Besides web agents, mobile GUI agents are gaining more and more popularity, which are developed to handle intricate tasks automatically on digital devices like smartphones. ERICA (Deka et al., 2016) defines a system for interaction mining in Android applications. It employs a human-computer interaction approach to capture the data, making it scalable and capable of capturing a wide range of interactions. PIXEL (Rust et al., 2023) and Pix2Struct (Lee et al., 2023) show promising capability in multilingual transfer and UI navigation, respectively, but they struggle with language understanding tasks compared to text-only LMs like BERT, limiting their utility. Patch-and-Text Prediction (PTP) proposed in (Gao et al., 2024) leads to better language understanding capabilities by masking and recovering both image patches and text within screenshots; AppAgent (Zhang et al., 2025) presents a multimodal agent that operates smartphone apps through low-

level actions like tapping and swiping, mimicking human interactions. The agent learns app functionalities through exploration, either autonomously or by observing human demonstrations, and then applies this knowledge to execute tasks; Mobile-Agent (Wang et al., 2024), which uses visual perception tools to locate operations on a mobile device using screenshots. It involves OCR models for identifying visual and textual elements, while realizing self-planning and self-reflection to autonomously navigate mobile apps, with a benchmark called Mobile-Eval introduced for performance evaluation; SeeClick (Cheng et al., 2024) is a visual GUI agent that operates solely on screenshots, bypassing the need for structured text. It employs Large Vision-Language Models (LVLMs) enhanced with GUI grounding pre-training to accurately locate screen elements based on instructions. The method involves automating the curation of GUI grounding data and creating a GUI grounding benchmark named ScreenSpot. It adapts universally to various GUI platforms and relies on screenshots. Simplifying the action space to clicking and typing.

C.1 Other Benchmarks

Shi et al. (2017) and Liu et al. (2018) establish a platform of website widgets where agents can complete online tasks through basic keyboard and mouse operations. APIBench (Li et al., 2023b), introduced by Gorilla(Patil et al., 2024), is a tool-augmented LLM benchmark to assess the tool utilization abilities of agents for code generation tasks. AgentBench (Liu et al., 2024) steps forward to provide a more general toolkit with lots of closed-box environments to assess agents' performances in answering user queries.

Webshop (Yao et al., 2022) builds a simulated e-commerce environment featuring 1.18 million real-world products, complemented by 12,087 crowd-sourced textual instructions. It converted the action prediction task into a choice-based imitation learning process, which facilitated the accuracy and ability for task execution. However, this approach failed to evaluate open-ended agent actions in the real world. It was also limited by its monotonous design of a one-website environment, which resulted in only a single category of web browsing tasks. Mind2Web (Deng et al., 2023) tries to construct a generalist web agent, which creates a dataset for crafting and benchmarking web agents

by the ability of instruction following. It proposed a two-stage training to convert the action prediction problem into MCQs. SeeAct (Zheng et al., 2024) is a following work that enabled multimodal information for visually understanding rendered web-pages and generating more accurate action plans. WebVoyager (He et al., 2024) is capable of capturing screenshots of web pages and then using JavaScript tools to automatically identify interactive elements based on the types of webpage elements. WebArena (Zhou et al., 2024) deploys a standalone set of multicategory websites in an interactive environment. VisualWebArena (Koh et al., 2024) is a subsequent project that built upon the foundation of WebArena, introducing the reliance on visual cues into the benchmark's design. The tasks of existing benchmarks are oversimplified whose completions requiring a single website, which is highly diverged from the natural web browsing tasks and should originally be designed for multihop over a long-horizon setting.