RealWebAssist: A Benchmark for Long-Horizon Web Assistance with Real-World Users

Suyu Ye*[1], Haojun Shi*[1], Darren Shih [1], Hyokun Yun [2], Tanya G. Roosta [2], Tianmin Shu [1]

[1]Johns Hopkins University,
[2]Amazon.com
{sye10, hshi33, dshih5, tianmin.shu}@jhu.edu, {yunhyoku,troosta}@amazon.com

Abstract

To achieve successful assistance with long-horizon web-based tasks, AI agents must be able to sequentially follow real-world user instructions over a long period. Unlike existing web-based agent benchmarks, sequential instruction following in the real world poses significant challenges beyond performing a single, clearly defined task. For instance, real-world human instructions can be ambiguous, require different levels of AI assistance, and may evolve over time, reflecting changes in the user's mental state. To address this gap, we introduce RealWebAssist, a novel benchmark designed to evaluate sequential instruction-following in realistic scenarios involving long-horizon interactions with the web, visual GUI grounding, and understanding ambiguous real-world user instructions. RealWebAssist includes a dataset of sequential instructions collected from real-world human users. Each user instructs a web-based assistant to perform a series of tasks on multiple websites. A successful agent must reason about the true intent behind each instruction, keep track of the mental state of the user, understand user-specific routines, and ground the intended tasks to actions on the correct GUI elements. Our experimental results show that state-of-the-art models struggle to understand and ground user instructions, posing critical challenges in following real-world user instructions for long-horizon web assistance.

Introduction

As an integral part of people's daily life, many of our everyday tasks are performed on the internet. With the tremendous advances in open-ended agents driven by large reasoning models (LRMs) and vision-language models (VLMs), there has been increasing interest in engineering web-based agents that can assist humans with complex tasks on the web following humans' instructions (Zheng et al. 2024a; Nakano et al. 2022). Recent works have demonstrated the promising performance of web-based agents on planning (Putta et al. 2024; Wang et al. 2024; Yao et al. 2023) and Graphical User Interface (GUI) grounding (Cheng et al. 2024; Wu et al. 2024b; Gou et al. 2024; Yang et al. 2024; Xu et al. 2024), across diverse websites, tasks, and GUI interfaces.

Despite these encouraging results, there have not been systematic studies on long-horizon web assistance with real-world users. Existing benchmarks (e.g., (Zhou et al. 2023; Deng et al. 2024; Cheng et al. 2024; Yao et al. 2022; Jang et al. 2024)) typically focus on performing a task based on a single instruction. Additionally, the instructions in the current benchmarks were not collected from real users during natural web use sessions, lacking the realism of real user instructions. As a result, these benchmarks do not capture the full complexity of real users' web behavior and instructions.

To bridge this gap, we propose **RealWebAssist**, the first sequential instruction following benchmark that evaluates long-horizon web assistance with real-world users. As illustrated in Figure 1, to perform a task, a user will instruct an AI assistant in a long sequence. Based on the past instructions and screenshots, the AI assistant must execute one or a few steps of actions to perform the latest instruction. Additionally, a user can engage in repeated interactions over a series of tasks with the assistant in a long session up to 40 minutes. To construct RealWebAssist, we recruited real users to instruct an assistant to perform multiple real-world tasks on the web. We created a large dataset with real user instructions (in both speech and text) for diverse real-world tasks and websites (as shown in Figure 2).

The sequential instruction following tasks in our RealWebAssist benchmark reflect the natural human behavior on the web. First, real-world users may not initially know what they are looking for. Thus, they need to engage in information seeking on multiple web pages (e.g., step 1-2 in Figure 1), sometimes even across websites. Second, based on new information such as product reviews, users may change their minds (e.g., step 3). Third, users give simple instructions that are seemingly ambiguous out of the context but could be interpreted based on spatial and temporal context via pragmatic reasoning (Goodman and Frank 2016; Fried et al. 2023). For instance, the third instruction in Figure 1 does not explicitly describe which product, but an intelligent assistant should be able to infer the true user intent and correctly select the product in the user's mind. Lastly, in our benchmark, users can browse the websites and have the autonomy to make critical decisions (such as purchasing) on their own, which is complementary to existing benchmarks that focus on agents' planning ability to fully complete the tasks without human involvement.

We systematically evaluate state-of-the-art models, including GUI grounding, VLMs, and large reasoning mod-

*These authors contributed equally.

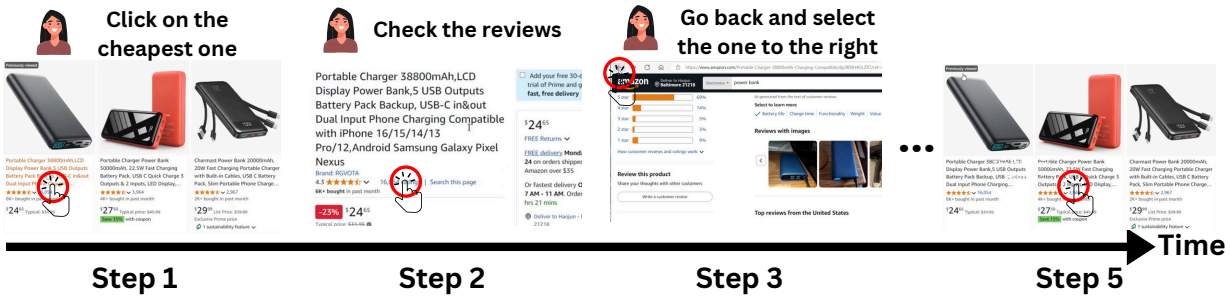

Figure 1: An example sequential instruction following task with a real-world user. The red circles indicate the correct actions based on the user's spoken instructions. Sequential instructions introduce unique challenges, such as the need to retain and reason over past context. For instance, the instruction in step 3 requires information from step 1 to be correctly interpreted.

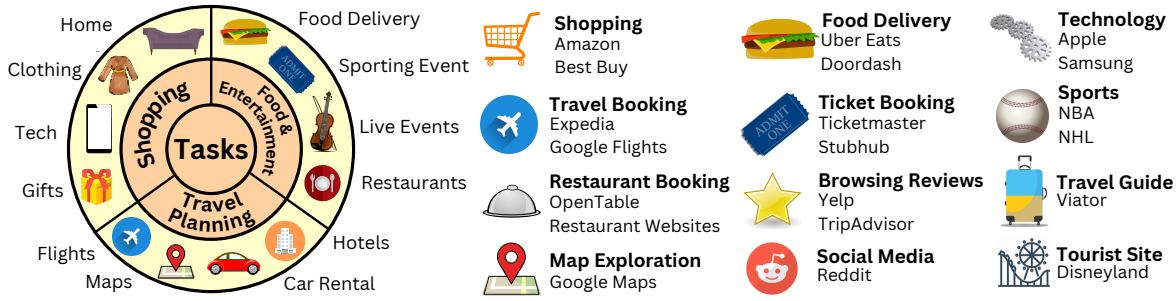

Figure 2: Examples of general task categories (left) and websites visited (right) in RealWebAssist. The tasks span a wide range of real-world scenarios, from shopping to food & entertainment to travel planning, which encourages users to visit many different websites.

els. Experimental results reveal that these models lack several key abilities, including grounding, understanding user intents, reasoning about spatial and temporal context, and adapting to user-specific routines.

Related Works

Web Agent Benchmarks. Existing web agent benchmarks primarily evaluate the performance of web agents on tasks with clearly defined, unambiguous instructions, often overlooking the complexities of real-world users' behavior and their instructions to an AI assistant. On WebArena (Zhou et al. 2023), Mind2Web (Deng et al. 2024), and WebShop (Yao et al. 2022), an agent follows a single instruction to perform an isolated task. While they offer an evaluation of an agent's planning capacity, they lack the evaluation of an agent's ability to follow a long sequence of user instructions on long-horizon web tasks. There have also been GUI grounding benchmarks, such as ScreenSpot (Cheng et al. 2024), that focused on grounding simple instructions to clicking actions on webpages. These instructions only instruct web agents to click web elements rather than reaching a user goal (e.g., purchasing an item). WebLINX (Lù, Kasner, and Reddy 2024) features sequential instruction following. However, the instructions were generated by annotators who received detailed guidelines and extensive training, rather than by actual users. The resulting instructions do not capture the nuances and complexity of real-world user instructions that naturally emerge in interactions with an assistant. In contrast, RealWebAssist consists of sequential instruction following tasks for assisting real-world users, providing a novel set of challenges necessary for long-horizon web assistance for real-world users. Table 1 summarizes key differences between RealWebAssist and prior benchmarks.

Autonomous Web Agents. There have been many recent works on engineering autonomous web agents through retrieval augmented planning (Kim et al. 2024; Zhou et al. 2024; Wu et al. 2024a; He et al. 2024; Pan et al. 2024), finetuning (Hong et al. 2024; Gur et al. 2024; Deng et al. 2024; Pang et al. 2024; Zhang and Zhang 2024), learning workflows (Zhang et al. 2023; Wang et al. 2024; Zheng et al. 2024b; Majumder et al. 2023; Cai et al. 2024), reinforcement learning (Liu et al. 2018; Shi et al. 2017; Nogueira and Cho 2016; Humphreys et al. 2022), and combinations of these methods (Liu et al. 2023; Putta et al. 2024). These works focus on planning for a single task. However, there has not been much work on understanding and following real-world users' sequential instructions on long-horizon tasks.

GUI Grounding. One key ability for web agents in many assistance tasks is to ground instructions to clicking actions on a webpage. Recent works have explored VLM finetuning (e.g., (Gou et al. 2024; Wu et al. 2024b; Yang et al. 2024, 2025; Wu et al. 2025; Qin et al. 2025; Xu et al. 2025; Yuan et al. 2025)) as well as prompting pretrained VLMs with segmentations of web elements (e.g., (Yang et al. 2023)) for enabling GUI grounding. These methods generate coordinates or bounding boxes on webpages to indicate where to click.

Benchmark	Real User	Sequential Instructions	Real Websites	GUI Grounding	Speech	# Instructions
SreenSpot (Cheng et al. 2024)	✗	✗	✓	✓	✗	1200+
WebArena (Zhou et al. 2023)	✗	✗	✗	✗	✗	812
Mind2Web (Deng et al. 2024)	✗	✗	✓	✗	✗	2000+
WebLINX (Lù, Kasner, and Reddy 2024)	✗	✓	✓	✗	✗	512
VideoWebArena (Jang et al. 2024)	✗	✗	✗	✗	✓	2021
WebShop (Yao et al. 2022)	✗	✗	✗	✗	✗	12087
BearCubs (Song et al. 2025)	✗	✗	✓	✗	✗	111
RealWebAssist (Ours)	✓	✓	✓	✓	✓	1885

Table 1: Comparison between RealWebAssist and existing web agent benchmarks on several key aspects: (1) whether instructions were given by real-world users instead of annotators, (2) whether there is a sequence of instructions, (3) whether there are real-world websites, (4) whether the agent needs to execute actions by selecting coordinates on webpages, (5) whether the instructions are speech instructions, and (6) the number of total instructions.

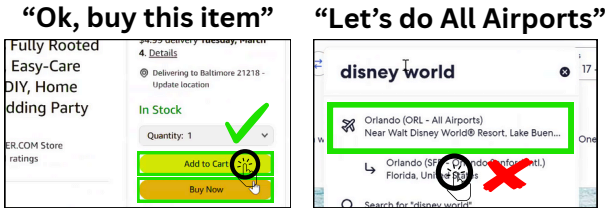

Figure 3: Multiple actions can satisfy a user's intent. A web agent's action is considered correct if the coordinate they provide is within one of the annotated correct regions.

They have only been trained on low-level instructions that clearly refer to web elements. It remains unclear if they can understand real-world user instructions that must be interpreted considering context or may refer to high-level goals.

RealWebAssist Benchmark

Problem Setup

RealWebAssist evaluates agents' ability to follow long-horizon, sequential web instructions to assist users with their high-level goals. In each task, a human user will try to reach an open-ended goal such as "buy formal outfits for a formal event" by instructing the assistant through a series of spoken instructions. The dataset is collected from interactions between human users and human assistants in a human experiment. To evaluate agents, we use the human assistants' actions to evaluate the agents' success.

In RealWebAssist, a web agent has access to the current instruction, webpage (as a screenshot), and all the past interactions (previous instructions & screenshots of webpages). Since we are focusing on tasks on real-world websites, it is challenging to ensure safety and reproducibility in an interactive evaluation setting. Therefore, we adopt an offline evaluation setting following prior web-based agent benchmarks with real websites (Deng et al. 2024; Cheng et al. 2024). Specifically, for each instruction collected from the human experiment, the agent needs to identify the correct element to interact with by providing a coordinate or a bounding box to click on the webpage. As shown by figure 3, a web agent's action is considered correct if the coordinate or the center of the bounding box they provide falls in the annotated correct regions on the webpage. If there are multiple steps corresponding to one instruction, we evaluate if the web agent's actions for the same instruction are all correct.

Evaluation Metrics

We consider the following evaluation metrics:

- **Task success rate:** A task is successful if the web agent can correctly produce actions for all instructions in a task.
- **Average progress:** We measure the progress of a task by the percentage of consecutive instructions the web agent can successfully perform before its first error in the task.
- **Step success rate:** We also consider a teacher forcing setting as a simpler, diagnostic evaluation, where the web agent will only need to follow the instruction at a single step of a task assuming all previous instructions have been successfully performed.

Dataset Construction

Setup. We recruited 10 participants (4 female, 6 male, mean age = 20 years) from a US university campus, none of whom had prior knowledge of the study's purpose, to construct the dataset. All participants were native or fluent English speakers. Each participant completed a 40-minute real-world web assistance session in which they tackled a series of open-ended tasks designed to encourage diverse strategies. During each session, participants verbally instructed an experimenter, who operated the computer on their behalf, to complete the tasks. We captured screen recordings and used a high-quality USB microphone to record speech as raw data. The user study was approved by an institutional review board.

User Tasks. To increase the instruction diversity and realism, participants received general web-based tasks requiring active information seeking, sub-goal planning, and comparison among various options. We generated the task list by few-shot prompting GPT-4o with open-ended tasks, followed by manual filtering and editing to ensure task quality and feasibility. These tasks provide only general guidance,

ensuring flexibility for personal decision-making. Example tasks include "Purchase an outfit for a formal event" and "Plan a 5-day trip to Japan, booking both flights and hotels". Each user finishes about 10 tasks.

Emergent User Behavior. In our realistic, open-ended settings, users exhibit rich behaviors that are not present in previous benchmarks. These include, but are not limited to, information seeking, researching and comparing different options, change of mind, and trial-and-error.

Annotations. We manually labeled RealWebAssist data to ensure high-quality annotations. We first segmented the full recording into individual clips corresponding to each user's instructions. In our benchmark, we disregard user speech unrelated to explicit instructions for the assistant, such as filler words or verbalized thought processes. For each instruction, we provide raw speech, speech transcript, webpage, and the correct regions to click (in the form of one or more bounding boxes). When there were multiple correct answers for the instructions (for instance, "can you close all the current tabs"), we annotated all correct regions with multiple bounding boxes. When the experimenter made a mistake during the data collection sessions, we annotated the correct action intended by the user. If an instruction required multiple steps to complete, we set the instruction at each step as the same instruction. To generate the text instructions, we used an off-the-shelf recognition model, Whisper Large-V3 (Radford et al. 2023), to transcribe users' speech and then manually fixed transcription errors. For all the instructions, we have three annotators verifying all of them, ensuring 100% agreement.

Dataset Statistics. RealWebAssist contains 1,885 user instructions across 107 tasks, 66 websites, and 2,524 screenshots. In addition to the benchmark, we also plan to release the raw data, consisting of over 6 hours of video & audio.

Key Challenges

RealWebAssist features multiple challenges as illustrated in Figure 4, including spatial and temporal reasoning needed to understand ambiguous and context-dependent user instructions, planning for multiple steps of actions to reach the goal communicated by an instruction, and learning about user-specific routines. These key challenges provide a more realistic and holistic evaluation of a web agent's reasoning, planning, and learning abilities to assist real-world users on long-horizon tasks. It is worth noting that many of these challenges, in particular, spatial reasoning, temporal reasoning, and routine understanding, are not present in existing web agent benchmarks. Unlike RealWebAssist, prior benchmarks, such as ScreenSpot (Cheng et al. 2024), WebArena (Zhou et al. 2023), and Mind2Web (Deng et al. 2024), only include clear, unambiguous, and non-sequential instructions.

Spatial Reasoning. When referring to one of the elements on a webpage, real-world users tend to use a concise instruction that can be understood conditioned on spatial context instead of an overly elaborated instruction. For instance, when instructing an assistant to buy a product, users may give short instructions such as "select the cheapest one," instead of describing the desired product in detail. Figure 4A depicts different types of spatial reasoning that rely on diverse spatial contexts, including ranking, spatial relations, and overall website functionalities. It is worth noting that these instructions may sometimes reveal users' preferences (e.g., preferred seating), providing additional information for the web agent to provide potentially more customized assistance in the future.

Temporal Reasoning. In our sequential instruction following tasks, users may instruct an assistant with the history as an assumed temporal context. For example, to understand the intended meaning of "click the last item," the assistant must memorize the items the user has viewed in the past. Figure 4B shows temporal reasoning based on different kinds of temporal context, ranging from short context between two consecutive webpages to long context with the same website to long context across websites. From the temporal context, the assistant needs to memorize crucial elements in the previous webpages, infer and track a user's mind (e.g., change of mind about what to buy) based on the past instructions and webpages, and identify the earlier webpage the user refers to. Such temporal reasoning has not been evaluated in prior web agent benchmarks. However, it is very common in our benchmark due to the nature of human web browsing behavior as well as human instructions guided by pragmatics (Goodman and Frank 2016).

Multi-step Planning. Many instructions require multiple steps to complete. In these cases, the assistant needs to interpret the goal implied by the instruction and plan a sequence of actions to achieve that goal. This goes beyond grounding the instruction to a single action on the current webpage. Figure 4C shows an example where the agent was asked to repeat the same order on another food delivery website to check if the price would be different. A successful execution of this instruction would require the agent to first understand what the order is to ground the goal on the current website and generate a successful multi-step plan.

Routine. Since our benchmark allows a user to engage in repeated interactions with an assistant over multiple tasks, we observe that users may define routines understood by the assistant after repeated interactions. As shown in Figure 4D, the user initially gave detailed step-by-step instructions when selecting arrival and departure dates for a flight. In a subsequent task, however, the user simplified them into a single instruction when selecting dates for a hotel room. Such shorter instructions become possible after establishing a routine in the earlier task. Cognitive studies found that procedural abstraction, like these routines, naturally emerges in human cooperative communication through repeated interactions, allowing more efficient communication with partners (McCarthy et al. 2021). The emergence of such routines in our benchmark poses a novel challenge for web agents—learning user-specific procedural abstraction via repeated interactions to achieve human-like adaptive assistance. We hypothesize that this ability could enhance users' perception of the AI assistant, as it understands human cooperative communication.

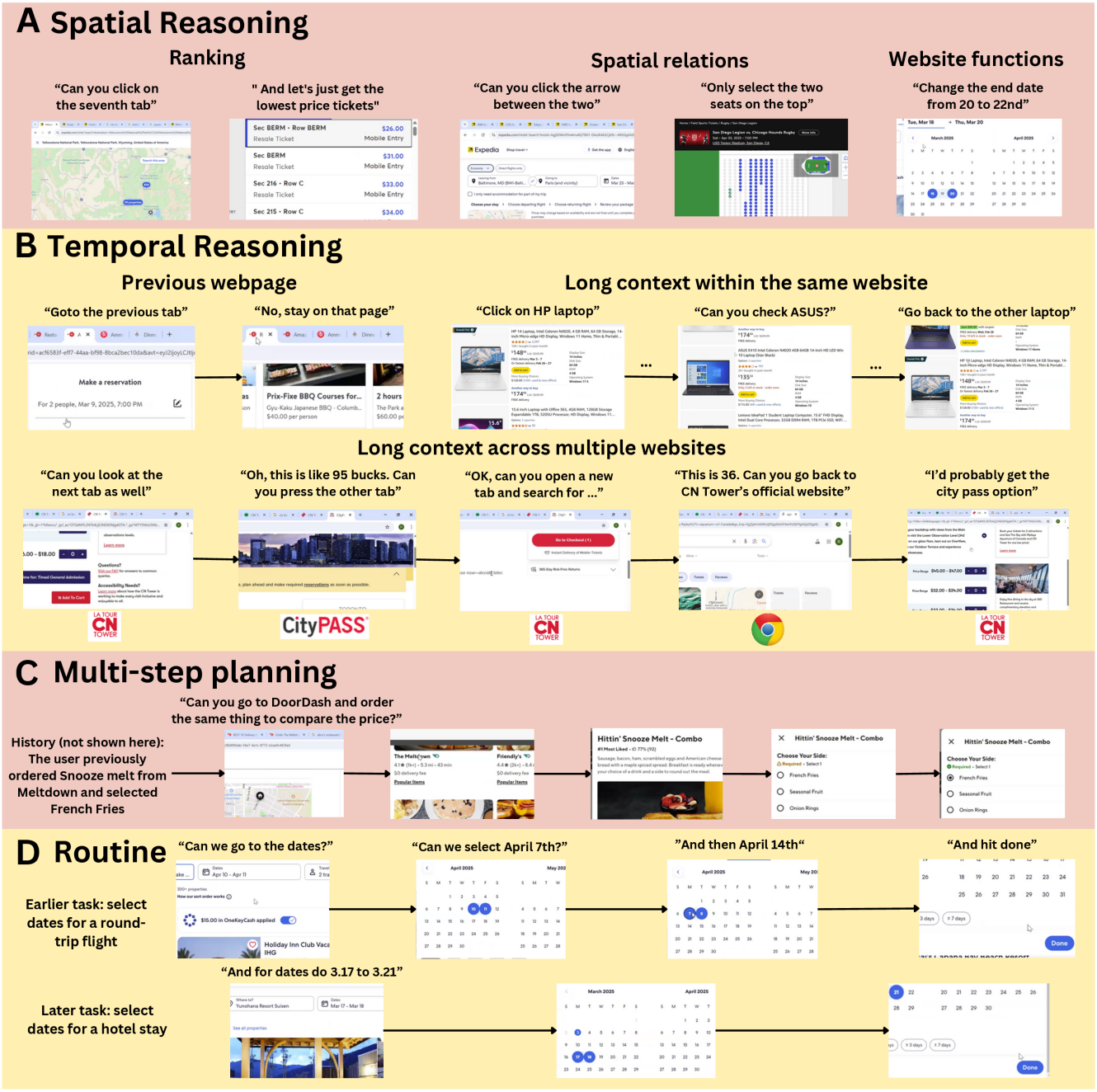

Figure 4: Key challenges introduced by RealWebAssist: (A) spatial reasoning, (B) temporal reasoning, (C) multi-step planning, and (D) learning user-specific routines.

Experiments

Baselines

We evaluated several types of models for web agents commonly evaluated in existing web agent benchmarks that have real-world websites (i.e., offline evaluation). For all the experiments, we use the ground-truth captions for instructions.

GUI Grounding Models. GUI grounding models directly translate an instruction to an action on a webpage. There are two general types of grounding models. First, Set-of-Mark (SoM) (Yang et al. 2023) segments salient elements on a webpage using an off-the-shelf segmentation model (e.g., SAM (Kirillov et al. 2023) and Semantic-SAM (Li et al. 2023)) and prompts a VLM to select a segment mask to identify the clicking area corresponding to the given instruction. Second, VLMs finetuned on datasets with paired instructions and annotated clicking coordinates or bounding

boxes. We evaluated UGround-V1 (Gou et al. 2024), OS-Atlas (Wu et al. 2024b), Aria-UI (Yang et al. 2024), GTA-1 (Yang et al. 2025), GUI-Actor (Wu et al. 2024a), and UI-TARS (Qin et al. 2025).

VLM/LRM + Grounding. Grounding models are designed or trained to ground a simple instruction to a webpage and thus tend to lack reasoning or planning capabilities. To address this, we leveraged VLMs and LRMs to first translate real user instructions to more understandable ones for grounding models. In particular, a VLM or an LRM needs to reason about the true user intent implied by the instruction and the spatial & temporal context. For instructions that require multiple actions, it needs to generate a plan to complete the instructions. Finally, it needs to generate a straightforward, clear instruction for the grounding model to produce the final action at each step. We evaluated state-of-the-art VLMs (OpenAI 2023; Team 2025; Qwen et al. 2025), as well as state-of-the-art LRMs (Jaech et al. 2024; Team 2025; Anthropic 2025). In the main results, we paired each VLM and LRM with the grounding model that achieved the highest step accuracy (GTA-1). For all VLMs and LRMs, we provide the past 10 steps for context, which we found to be a reasonable fixed context length in our preliminary study, balancing cost and informativeness. We also found that prompting models with screenshots of past webpages could incur a high cost. Therefore, we only prompt the models with the screenshot of the current webpage. For the history, we prompted GPT-4o to generate text-based action history based on consecutive screenshots and the instructions at each step. We then used this text-based history description for the evaluated VLMs and LRMs.

Finetuning. To evaluate whether models can learn to better follow real-world user instructions with additional training, we finetuned the best-performing grounding model (GTA-1) following the model's original group relative policy optimization (GRPO) training procedure (Yang et al. 2025) on 9 participants' data and tested it on the held-out participants' instructions. Specifically, we trained the grounding model to produce an action based on the past 10 steps of actions (in text), the current webpage screenshot, and the instruction. We enumerated different train/test splits and reported the averaged performance, either using the finetuned model alone or pairing it with the best VLM or LRM.

Results

Main results are summarized in Table 2. All models fell short in following real user instructions. The highest task success rate was only 14.0%, and the highest average progress was only 28.7%, a large gap compared to humans (93.4% task success rate). This difference has a 95% confidence interval of [71.3, 87.5], and is highly significant with p-value < 0.0001. Grounding methods by themselves failed to finish most tasks. However, when paired with the best-performing grounding model (GTA-1), instructions generated by VLMs & LRMs significantly improved the performance. LRMs performed marginally better than most VLMs. Across all three metrics, Gemini 2.5 Flash, Gemini 2.5 Pro, and o3 showed the strongest performance. Finetuning GTA-1 on real user data marginally improved its performance, but finetuning offered no benefit when GTA-1 was paired with VLMs and LRMs, since the finetuned model is trained to adapt to real users' instructions instead of instructions generated by VLM or LRM.

Discussion

Can grounding models understand real-world user instructions? There remains a significant gap in the performance of current direct grounding methods. The best grounding model, GUI-Actor, has a task success rate of only 5.7%. Figure 5 illustrates various failure cases encountered when directly using GTA-1. Unsurprisingly, grounding models fail to interpret instructions requiring reasoning due to their limited reasoning capabilities. However, even for context-free instructions involving straightforward spatial reasoning—tasks where grounding methods should excel—they frequently misinterpret spatial layouts or rankings. For instance, they often incorrectly select elements for instructions such as "click the first one."

How can VLMs & LRMs help? VLMs or LRMs can convert the original user instructions into more direct and explicit descriptions that a grounding model can more easily understand. This is made possible by their reasoning capacities. For instance, in Figure 5A, the grounding model (GTA-1) on its own fails to select the first tab: it selects the first element instead of the first tab. However, it succeeds after o3 rewrites the instruction to refer to the title. As shown in Figure 5B, grounding models may sometimes still fail due to inherent limitations even when VLMs/LRMs generate clearer instructions. Nonetheless, incorporating VLMs or LRMs significantly improves overall performance.

What are the limitations of VLMs & LRMs? While VLMs and LRMs help, the highest task success rate is still only 14.0%. Beyond errors from grounding models (e.g., Figure 5B), they continue to struggle with complex temporal reasoning. In Figure 5C, the user previously asked to open the first two search results in new tabs. When later instructed to "look at the first one we just opened," o3 failed to identify which element "the first one" referred to—instead of the first newly opened tab, it pointed to the first search result. We further analyze the error distribution between reasoning errors (the VLM/LRM mistranslates the instruction and refers to the wrong element) and grounding errors (the rewritten instruction is correct, but the grounding model still fails to click the right element). For the best model (o3 + GTA-1), 43.3% of errors are grounding errors and 56.7% are reasoning errors. This suggests that current VLMs and LRMs still lack the reasoning and planning abilities needed to robustly perform sequential instruction-following tasks.

Does learning from real-world user data help? Finetuning GTA-1 marginally improved average progress and step accuracy but yielded no additional benefit when paired with VLMs and LRMs. These results show that the finetuned model better understands real user instructions, yet it still fails to generalize to instructions generated by VLMs and LRMs. The experiments suggest that finetuning grounding models on a small set of real user instructions provides minimal benefit, and collecting large-scale real user instructions remains a significant challenge.

Category	Model	Task Success	Progress	Step Accuracy
Human	Human Operator	93.4	96.4	99.2
Grounding	Set-of-Mark	0.0	2.7	29.8
	OS-Atlas	0.0	3.8	26.6
	Aria-UI	0.0	2.4	32.8
	UGround-V1	0.0	6.2	47.7
	UI-TARS	2.8	13.1	53.8
	GTA-1	3.7	17.7	61.5
	GUI-Actor	5.7	14.7	61.4
VLM + Grounding	GPT-4o + GTA-1	8.4	23.5	72.7
	Qwen 2.5 72B + GTA-1	9.3	24.3	69.0
	Gemini 2.5 Flash + GTA-1	11.2	26.9	75.4
LRM + Grounding	o1 + GTA-1	7.5	17.7	68.2
	Gemini 2.5 Pro + GTA-1	8.4	23.5	74.5
	o4-mini + GTA-1	10.3	21.7	67.1
	Claude 3.7 Sonnet + GTA-1	12.1	26.7	68.8
	o3 + GTA-1	**14.0**	**28.7**	**76.7**
Finetuned	GTA-1-F	3.7 (+0.0)	19.7 (+2.0)	64.3 (+2.8)
	Gemini 2.5 Flash + GTA-1-F	11.2 (+0.0)	26.9 (+0.0)	75.4 (+0.0)
	o3 + GTA-1-F	**14.0** (+0.0)	**28.7** (+0.0)	**76.7** (+0.0)

Table 2: Model Performance including task success rate, average progress, and step accuracy. All results are in %. The best performance of pretrained models and finetuned models is highlighted in bold. GTA-1-F indicates the finetuned GTA-1. Plus sign indicates the improvement compared to using the raw model for the same set of instructions.

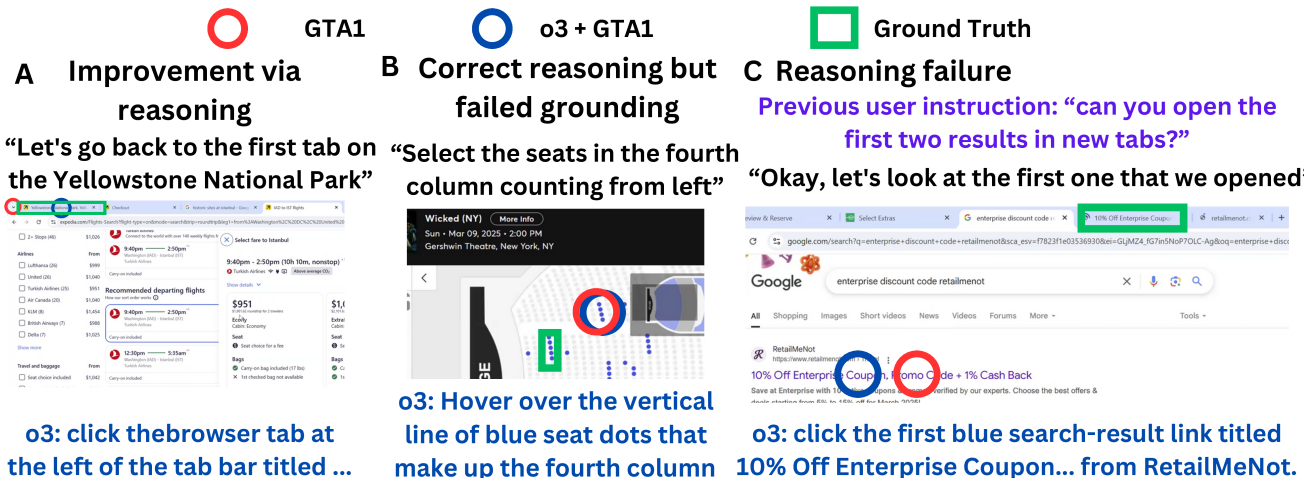

Figure 5: Qualitative results. The captions show instructions generated by o3 (the best LRM). (A) Error corrected by using o3 to convert instructions. (B) Failure caused by GTA-1 when o3 reasons correctly. (C) Reasoning failure caused by o3.

Limitations. RealWebAssist represents an important first step towards evaluating web agents on long-horizon, real-user tasks. However, it has several limitations. The first is participant scale and diversity. Collecting real-user data is expensive and time-consuming. The number of participants is comparable to prior works that use expert annotators (Lù, Kasner, and Reddy 2024). However, we intend to increase user diversity in future versions of the benchmark. We will also open-source our data collection tools for community expansion of the dataset. Second, like prior benchmarks on real-world websites (Deng et al. 2024; Cheng et al. 2024), we constrain our evaluation to an offline setting to ensure reproducibility and safety. This is complementary to benchmarks that focus on interactive evaluation in sandbox environments (e.g., WebArena). We believe that web agents should be evaluated on both types of benchmarks to fully assess their capabilities. Lastly, the current setting does not allow dialogue between a user and the AI assistant, which we will explore in future work.

Conclusion

In this paper, we present RealWebAssist, the first benchmark for evaluating web agents' ability to provide long-horizon web assistance with real-world users via sequential instruction-following. Our benchmark poses novel challenges, including spatial and temporal reasoning, planning, and adapting to user-specific routines. We conducted a comprehensive evaluation and analysis on multiple state-of-the-art GUI grounding models, VLMs, and LRMs, revealing critical limitations of them. We have also shown the limited benefit of finetuning models on real user data. Our benchmark, along with the well-annotated user instruction dataset, provides resources and diagnostic tools for further research on real-world web assistance. In future work, we plan to expand our human study to include more participants from various backgrounds, examine web assistance in interactive settings, and incorporate chat between users and web agents.

Acknowledgements

This work was supported by a research grant from Amazon. We thank Janice Chen for helpful discussions.

References

Anthropic. 2025. Claude 3.7 Sonnet and Claude Code. https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: 2025-03-17.

Cai, T.; Wang, X.; Ma, T.; Chen, X.; and Zhou, D. 2024. Large Language Models as Tool Makers. arXiv:2305.17126.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Fried, D.; Tomlin, N.; Hu, J.; Patel, R.; and Nematzadeh, A. 2023. Pragmatics in Language Grounding: Phenomena, Tasks, and Modeling Approaches. arXiv:2211.08371.

Goodman, N. D.; and Frank, M. C. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11): 818–829.

Gou, B.; Wang, R.; Zheng, B.; Xie, Y.; Chang, C.; Shu, Y.; Sun, H.; and Su, Y. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Gur, I.; Furuta, H.; Huang, A.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2024. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. arXiv:2307.12856.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. arXiv:2401.13919.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Zhang, Y.; Li, J.; Xu, B.; Dong, Y.; Ding, M.; and Tang, J. 2024. CogAgent: A Visual Language Model for GUI Agents. arXiv:2312.08914.

Humphreys, P. C.; Raposo, D.; Pohlen, T.; Thornton, G.; Chhaparia, R.; Muldal, A.; Abramson, J.; Georgiev, P.; Santoro, A.; and Lillicrap, T. 2022. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*, 9466–9482. PMLR.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jang, L.; Li, Y.; Zhao, D.; Ding, C.; Lin, J.; Liang, P. P.; Bonatti, R.; and Koishida, K. 2024. Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. *arXiv preprint arXiv:2410.19100*.

Kim, M.; Bursztyn, V.; Koh, E.; Guo, S.; and Hwang, S.-w. 2024. Rada: Retrieval-augmented web agent planning with llms. In *Findings of the Association for Computational Linguistics ACL 2024*, 13511–13525.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. arXiv:2304.02643.

Li, F.; Zhang, H.; Sun, P.; Zou, X.; Liu, S.; Yang, J.; Li, C.; Zhang, L.; and Gao, J. 2023. Semantic-SAM: Segment and Recognize Anything at Any Granularity. *arXiv preprint arXiv:2307.04767*.

Liu, E. Z.; Guu, K.; Pasupat, P.; Shi, T.; and Liang, P. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. *arXiv preprint arXiv:1802.08802*.

Liu, Z.; Yao, W.; Zhang, J.; Xue, L.; Heinecke, S.; Murthy, R.; Feng, Y.; Chen, Z.; Niebles, J. C.; Arpit, D.; et al. 2023. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *arXiv preprint arXiv:2308.05960*.

Lù, X. H.; Kasner, Z.; and Reddy, S. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

Majumder, B. P.; Mishra, B. D.; Jansen, P.; Tafjord, O.; Tandon, N.; Zhang, L.; Callison-Burch, C.; and Clark, P. 2023. CLIN: A Continually Learning Language Agent for Rapid Task Adaptation and Generalization. arXiv:2310.10134.

McCarthy, W. P.; Hawkins, R. D.; Wang, H.; Holdaway, C.; and Fan, J. E. 2021. Learning to communicate about shared procedural abstractions. *arXiv preprint arXiv:2107.00077*.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2022. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332.

Nogueira, R.; and Cho, K. 2016. End-to-end goal-driven web navigation. *Advances in neural information processing systems*, 29.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Pan, J.; Zhang, Y.; Tomlin, N.; Zhou, Y.; Levine, S.; and Suhr, A. 2024. Autonomous Evaluation and Refinement of Digital Agents. arXiv:2404.06474.

Pang, R. Y.; Yuan, W.; Cho, K.; He, H.; Sukhbaatar, S.; and Weston, J. 2024. Iterative Reasoning Preference Optimization. arXiv:2404.19733.

Putta, P.; Mills, E.; Garg, N.; Motwani, S.; Finn, C.; Garg, D.; and Rafailov, R. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*.

Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; et al. 2025. UI-TARS: Pioneering Automated GUI Interaction with Native Agents. *arXiv preprint arXiv:2501.12326*.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, 28492–28518. PMLR.

Shi, T.; Karpathy, A.; Fan, L.; Hernandez, J.; and Liang, P. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, 3135–3144. PMLR.

Song, Y.; Thai, K.; Pham, C. M.; Chang, Y.; Nadaf, M.; and Iyyer, M. 2025. Bearcubs: A benchmark for computer-using web agents. *arXiv preprint arXiv:2503.07919*.

Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. arXiv:2507.06261.

Wang, Z. Z.; Mao, J.; Fried, D.; and Neubig, G. 2024. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Wu, Q.; Cheng, K.; Yang, R.; Zhang, C.; Yang, J.; Jiang, H.; Mu, J.; Peng, B.; Qiao, B.; Tan, R.; et al. 2025. GUI-Actor: Coordinate-Free Visual Grounding for GUI Agents. *arXiv preprint arXiv:2506.03143*.

Wu, Z.; Han, C.; Ding, Z.; Weng, Z.; Liu, Z.; Yao, S.; Yu, T.; and Kong, L. 2024a. OS-Copilot: Towards Generalist Computer Agents with Self-Improvement. arXiv:2402.07456.

Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P. P.; et al. 2024b. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; and Xiong, C. 2024. Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction. arXiv:2412.04454.

Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; and Xiong, C. 2025. Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction. arXiv:2412.04454.

Yang, J.; Zhang, H.; Li, F.; Zou, X.; Li, C.; and Gao, J. 2023. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*.

Yang, Y.; Li, D.; Dai, Y.; Yang, Y.; Luo, Z.; Zhao, Z.; Hu, Z.; Huang, J.; Saha, A.; Chen, Z.; et al. 2025. GTA1: GUI Test-time Scaling Agent. *arXiv preprint arXiv:2507.05791*.

Yang, Y.; Wang, Y.; Li, D.; Luo, Z.; Chen, B.; Huang, C.; and Li, J. 2024. Aria-UI: Visual Grounding for GUI Instructions. *arXiv preprint arXiv:2412.16256*.

Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35: 20744–20757.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

Yuan, X.; Zhang, J.; Li, K.; Cai, Z.; Yao, L.; Chen, J.; Wang, E.; Hou, Q.; Chen, J.; Jiang, P.-T.; and Li, B. 2025. Enhancing Visual Grounding for GUI Agents via Self-Evolutionary Reinforcement Learning. arXiv:2505.12370.

Zhang, C.; Yang, Z.; Liu, J.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2023. AppAgent: Multimodal Agents as Smartphone Users. arXiv:2312.13771.

Zhang, Z.; and Zhang, A. 2024. You Only Look at Screens: Multimodal Chain-of-Action Agents. arXiv:2309.11436.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024a. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Zheng, L.; Wang, R.; Wang, X.; and An, B. 2024b. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. arXiv:2306.07863.

Zhou, A.; Yan, K.; Shlapentokh-Rothman, M.; Wang, H.; and Wang, Y.-X. 2024. Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models. arXiv:2310.04406.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.