Guiding the Blind: Generalizing GUI Agents to Unseen Websites via Multimodal Tutorials

Xinwei Long[1], Kai Tian[1⋆], Peng Xu[1], Weibo Gao[2], Yihua Shao[3], Guoli Jia[1], Haozhe Geng[4], Sa Yang[4], Jingxuan Li[5], Huayong Hu[5], Kaiyan Zhang[1,6], Jiaqi Wang[7], and Bowen Zhou[1,7] ✉

[1] Tsinghua University, Beijing, China
[2] University of Science and Technology of China, Hefei, China
[3] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[4] Peking University, Beijing, China
[5] Independent Researcher
[6] Horizon Research, Frontis.AI, Beijing, China
[7] Shanghai Artificial Intelligence Laboratory, Shanghai, China
Project Page: https://github.com/TsinghuaC3I/webOne
longxw22@mails.tsinghua.edu.cn, xiaotiantk@gmail.com

Abstract. Despite the remarkable progress of Large Multimodal Models (LMMs), deploying autonomous agents to navigate web Graphical User Interfaces (GUIs) remains a significant challenge. Most existing agents are "blind" when encountering unfamiliar websites, as they rely heavily on patterns memorized during in-domain training, which inevitably fails in the open-world web. To bridge this generalization gap, we propose that agents should mimic human behavior: leveraging external expertise to navigate unknown environments. In this paper, we introduce `WebOne`, a novel benchmark designed to evaluate an agent's ability to master unseen websites by referencing Multimodal Tutorials—heterogeneous knowledge sources derived from instructional videos, historical trajectories, and human demonstrations. `WebOne` comprises 1,342 real-world tasks and 970 high-quality tutorials across 60+ websites. Building upon this, we propose `WebLearner`, a reinforcement learning-based framework that utilizes a hierarchical referencing strategy to synthesize information from these multimodal tutorials. Experimental results demonstrate that `WebLearner` significantly outperforms current state-of-the-art open-source models, achieving a 56.9% success rate on held-out websites and showing competitive performance against recent proprietary models.

Keywords: GUI Agents · Large Multimodal Models · Benchmark

1 Introduction

Graphical User Interface (GUI) agents [4, 15, 22, 33, 45, 57, 64] have emerged as a novel human computer interaction paradigm across web [13, 25], mobile [41,

⋆ Kai Tian has equal contribution with Xinwei Long.

68, 80], and desktop platforms [11, 14, 30, 56]. Driven by the rapid evolution of LMMs [70], these agents assist humans by automating complex daily tasks. However, existing agents still face a generalization gap when deployed to unseen websites or encountering UI updates. In these unfamiliar environments, agents often struggle to establish reliable mappings between high-level user intents and localized web elements. This limitation frequently leads to action hallucinations or task failures in zero-shot scenarios.

To navigate unfamiliar interfaces, recent research has attempted to explore the potential of external knowledge sources [31], such as instructional videos [29, 81] or step-by-step blogs [71], to provide procedural guidance. Preliminary works in this area can be divided into two categories. The first [38, 78] focuses on learning during the training stage by using pseudo-trajectories extracted from internet resources. The second [71] involves scaling at test time, where agents are guided by external expertise to make correct decisions. Although these external resources provide essential procedural heuristics that transcend specific pixel values, effectively using such heterogeneous, sparse, and noisy external knowledge remains a significant challenge for automated agents. Furthermore, it remains unclear whether performance failures stem from insufficient external guidance or from the limited active learning capabilities of the agent itself.

In this paper, we argue that the key to overcoming the generalization bottleneck lies in the agent's ability to actively learn and follow external guidance. To this end, we introduce `WebOne`, a comprehensive and carefully curated Web GUI benchmark derived from 1342 online tasks across 60+ real-world websites. We divide `WebOne` into training and test sets by website, ensuring that websites and tasks in the test set are unseen during training. Moreover, we construct 970 high-quality text-vision interleaved tutorials from diverse sources, including human demonstrations, online tutorial videos, and execution records from proprietary models. To process these heterogeneous inputs, we develop an automated extraction pipeline that transforms them into a unified format for high scalability. Compared to the raw sources, these tutorials are more knowledge-rich and information-dense, offering hierarchical guidance that spans low-level UI elements, step-by-step instructions, and high-level global planning. Furthermore, leveraging our `WebOne` dataset, we benchmark 11 baseline models, including state-of-the-art proprietary models like Gemini 2.5 Pro and open-source LMMs.

Building on `WebOne` dataset, we aim to develop a GUI agent capable of autonomous learning and cross-website generalization. Our approach is inspired by the cognitive process of humans navigating unfamiliar interfaces, where they iteratively analyze the current state and retrieve relevant instructions from tutorials to make informed decisions. Drawing on this heuristic, we propose `WebLearner`, a reinforcement learning (RL)-based framework featuring a hierarchical memory system. It consists of a Short-term Observational Buffer for capturing immediate visual feedback, a Dynamic Task Progress Tracker for maintaining intra-task state consistency, and a Multimodal Knowledge Base for indexing and retrieving external tutorial insights. We optimize the model using step-level reward signals

to help it summarize useful information from both history and current states, refer to relevant tutorial steps, and execute correct actions. Experimental results on our `WebOne` dataset, as well as the open-source benchmarks `Mind2Web` [8] and `Video-WebArena` [23], demonstrate that WebLearner effectively extracts useful information from external tutorials to generalize to unseen websites. In contrast, in-domain SFT baselines struggle in unseen environments and often get stuck, while baselines that learn directly from videos fail to capture patterns relevant to the current task.

To our knowledge, our main contribution can be stated as:

- **Idea and dataset.** To our knowledge, we propose a multimodal tutorial-based idea to improve the test-time generalizability for web GUI agents. Moreover, we alleviate data scarcity by contributing `WebOne`, a web GUI dataset comprising 1342 online tasks and 970 high-quality multimodal tutorials sourcing from 60+ real-world and frequently-used websites.
- **Methodology and model.** We propose `WebLearner`, an RL-based agent model that learns from tutorials, reflects on the web interaction environment, and generates actions through reward-driven optimization.
- **Performance and potential.** We benchmark nine baselines on `WebOne`. Our `WebLearner` achieves state-of-the-art results against open-source baselines and competitive results compared to recent proprietary models.

2 Related Work

GUI agents represent a novel paradigm in human-computer interaction across desktop [24, 58, 69], mobile [9, 52, 54, 77], and web platforms [17, 23, 50, 62, 82]. It is worth emphasizing that GUI agents differ significantly from agents based on the MCP protocol [21]. While MCP-based web agents [10, 55, 60, 67, 76] typically invoke officially released APIs (e.g., search API) directly, GUI agents are required to interact with graphical interfaces through clicking and browsing, much like a human user. Earlier studies [20, 47] focused on specific software environments or operating system tasks by parsing structured files. Recent efforts [32, 51, 63, 66] aim to build general-purpose agents that can interpret raw visual signals and perform complex actions. With the advent of RLVR [79, 83], several works [40, 46, 65] employ RL for trial-and-error decision optimization, utilizing rule-based rewards [12, 44] or trajectory-level feedback [27, 39, 54, 61] to refine agent policies. While these models show strong performance in closed settings, they often fail when faced with unfamiliar layouts or dynamic interface changes. This limitation highlights the importance of Web GUI agents, which must navigate the open and constantly changing environment of the internet.

Web GUI Environments and Benchmarks. The progress of web GUI agents depends on diverse environments and evaluation benchmarks. Initial efforts [75] utilized oversimplified simulated environments, which were later replaced by static benchmarks [6, 8] that provided UI states and recorded actions. Although effective for evaluating step-level grounding, static settings often fail to capture

the interactive complexity of real-world websites. Consequently, recent benchmarks [26, 43, 50, 82] have transitioned to dynamic environments simulating live web interactions. Despite their success, existing benchmarks often contain outdated tasks [17] or assume access to in-domain demonstrations and focus on repetitive tasks within known sites [19], which struggle to evaluate how an agent generalizes to unseen websites using external guidance. This gap motivates our work on `WebOne`, which evaluates test-time adaptation in open-domain web scenarios.

Learning From External Source. Collecting real-world trajectories for GUI agents is expensive, leading researchers [29,72] to explore methods that utilize external knowledge. Existing methods can be categorized into two paradigms. The first involves agents learning from external sources during the training phase. For instance, TongUI [78] and GUICourse [5] generate training data from instructional videos, while SEAgent [59] distills computer use data from general LMMs. EvoCUA [73] focuses on iterative training using collected experience data. The second paradigm enhances agent capabilities during test-time through RAG [16,35–37,71] or tutorial-based methods. For instance, RAG-GUI [71] trains a simple summarizer to extract textual information from the Internet, with the agent model frozen. However, its model and data are not publicly available, which is hard to reproduce. Several mobile [31] and desktop [28, 29, 81] agents built on training free and api based frameworks attempt to perform imitation learning from instructional videos. Distinct from them, we use RL to train the agent to actively ground real-time observations into procedural heuristics in multimodal tutorials, which fosters a generalizable reference following capability across diverse websites instead of step-wise imitations.

3 `WebOne` Benchmark

Formulation. Web GUI agents are specialized agents that navigate, interpret, and interact with web-based graphical user interfaces. The decision-making process for such an agent and its operational environment are generally modeled within a partially observable Markov decision process (POMDP) framework [48] represented by a tuple $M = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, T(\cdot), R(\cdot) \rangle$. The notation is defined as follows: $\mathcal{S}$ is the state space, *i.e.*, the webpage states. $\mathcal{O}$ denotes the observation space including user intent, screenshots, *etc.* $\mathcal{A}$ is the action space that is a set of actions such as clicking, typing, and scrolling. $T(\cdot)$ works as the state transition function that maps a state-action pair to a subsequent state: $T(s, a) \rightarrow s'$. $R(\cdot)$ is termed the reward function that outputs a numerical signal at each step, providing immediate feedback to the agent's decision-making process: $r = R(s, a, s')$. To evaluate generalization, we split `WebOne` into training and test sets with disjoint websites, so that all test-set websites and totorials are unseen.

Multimodal Tutorial Construction. We curate multimodal tutorials from three primary sources: **(1)** Expert Demonstrations, **(2)** Instructional Videos from public platforms, and **(3)** Success Experiences from specialized agents. To convert these raw traces into high-level guidance, we use a "Capture-and-Refine"

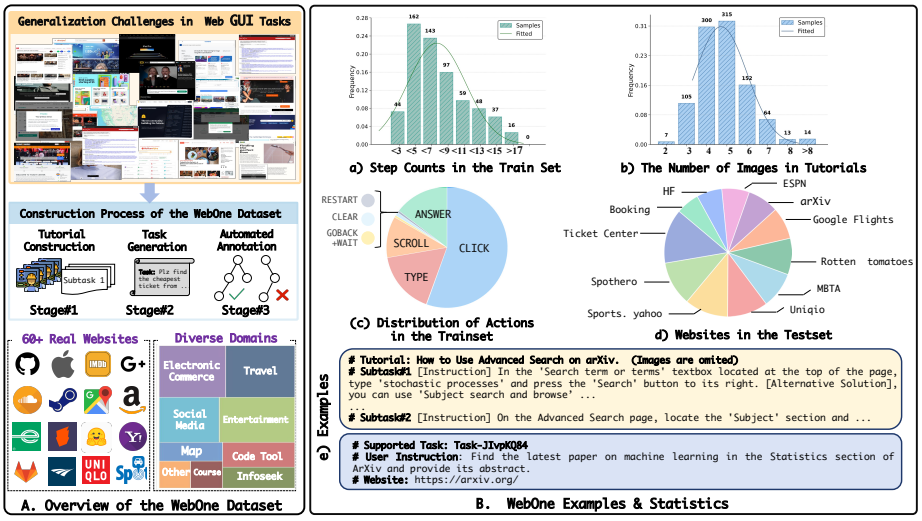

Fig. 1: Overview and statistics of `WebOne` dataset.

pipeline. First, redundant low-level events are filtered. Then, an LMM summarizes the remaining steps into a sequence of screenshots paired with procedural descriptions. This process ensures the tutorials capture the general logic of a website rather than specific, brittle action coordinates.

Task Construction. We design multiple filtering strategies to ensure our data quality. **(1)** In terms of difficulty, our samples are rigorously cleaned by filtering out outliers: a) impossible to complete, *e.g.*, *"scheduling a COVID-19 vaccine for the next week"*; b) overly simple, *e.g.*, *"finding a website's help page"*; or c) excessively complex (requiring > 20 atomic actions). **(2)** For time-sensitive tasks, we verify their feasibility. For example, it is impossible to purchase past-dated flight tickets. Absolute timestamps are replaced with relative expressions to maintain task validity over time, such as changing *"buy a ticket for October 23, 2024"* to *"buy a ticket for the 23rd of next month"*. Moreover, to further increase the task diversity, we collect seed tasks by sampling and rewriting existing ones. These seeds are then used as in-context examples to prompt an LLM to generate novel tasks. Finally, all generated tasks are manually reviewed and refined to ensure their uniqueness and feasibility.

Training Trajectory Collection. We automate our annotation by designing a multi-agent framework with three components that are initialized with the strong proprietary model [7]. **(1)** Planner: Comprehends the multimodal tutorial to generate a step-by-step plan. **(2)** Executor: Learns low-level information in the tutorial and produces the next action according to the plan. **(3)** Verifier: Assesses the action outcomes. These components operate iteratively until the Planner identifies a termination state. The resulting trajectories are categorized into three predefined types: unfinished (exceeded max iterations), finished but unsuccessful, or finished and successful. We retain only the successful tra-

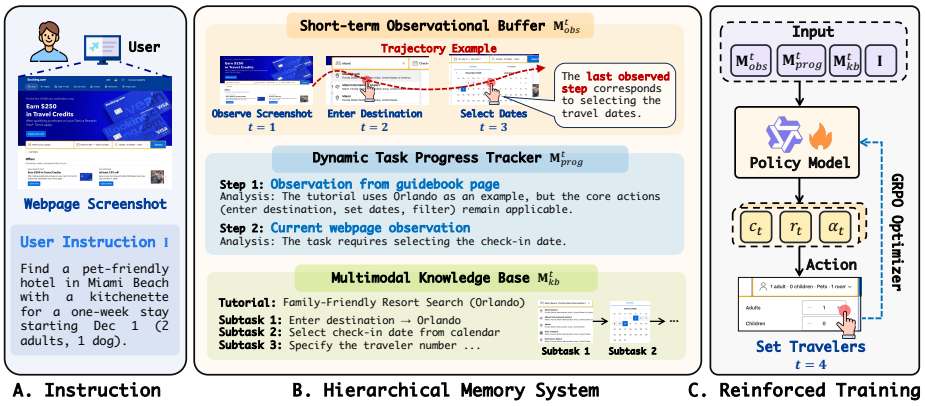

Fig. 2: Methodology overview of our `WebLearner`: (A) The agent receives a user instruction and the current webpage observation as input. (B) A hierarchical memory system integrates short-term observations, task progress tracking, and multimodal tutorial knowledge. (C) The policy model generates analysis, reference, and action outputs, which are optimized via GRPO-based reinforcement learning.

jectories, preserving the complete sequence of observations and actions while discarding all intermediate outputs.

Benchmark Statistics. As shown in Fig. 1, `WebOne` spans nine distinct domains, including E-comm., Social Media, Travel, and so on. The dataset is partitioned into a training set of 637 tasks and a held-out test set of 705 tasks. The latter involves websites the agent never encountered during training, thereby requiring it to interpret and follow tutorials to navigate novel UI structures. All trajectory data were collected automatically, with most trajectories consisting of five to seven atomic actions. In terms of action distribution, click actions account for over 50% of the total, while type, scroll, and answer actions each constitute approximately 15%. The remaining 5% consists of recovery actions, such as wait, go-back, and restart. Regarding the multimodal tutorials, we constructed 0.9K high-quality samples, most of which contain four to five screenshots. Detailed annotation protocols and per-category distributions are provided in the Appendix.

4 Methodology

Overview. We propose `WebLearner`, an RL-driven hierarchical memory framework that enables web GUI agents to learn, adapt, and generalize to unfamiliar websites at test time. It is designed to tackle the specific challenges in web GUI tasks and leverage the distinctive properties of our `WebOne` dataset. The model, depicted in Figure 2, operates end-to-end. In an interactive loop, it takes its current memory state to produce a triple of $\langle thinking, reference, action \rangle$. The web engine executes the action, yielding a new observation (*e.g.*, new pages) and feedback (*e.g.*, successful messages). The memory is then updated with this new

information, and the process iterates until termination. Step-wise performance is guided by a hybrid rule-based and model-based reward function. The resulting rewards directly drive the policy updates.

Hierarchical Memory System. We introduce a hierarchical memory system to handle long, interleaved multimodal contexts. The system comprises three levels of specialized memory storage:

(1) Short-term Observational Buffer. The Short term Observational Buffer ($\mathcal{M}_{obs}$) functions as the agent's immediate sensory organ. It stores the raw visual inputs and structured semantic metadata from the most recent k interaction steps. Unlike static inputs, this buffer allows the agent to perceive temporal changes in the UI, such as pop up windows or loading states, thereby providing a stabilized observation space for action prediction.

(2) Dynamic Task Progress Tracker. To avoid the repetitive behavior common in large model based agents, the Dynamic Task Progress Tracker ($\mathcal{M}_{prog}$) maintains a running summary of completed sub goals and failed attempts within the current episode. At each step t, the agent synthesizes the current state s_t and previous action a_{t-1} into a logical progress state. This tracker ensures that the agent maintains a coherent trajectory towards the final objective, especially when the target website exhibits complex multi page navigation logic.

(3) Multimodal Knowledge Base. The Multimodal Knowledge Base ($\mathcal{M}_{kb}$) serves as an externalized repository of cross-site heuristics, consisting of image-text interleaved tutorials that demonstrate generalized procedures. To effectively leverage this knowledge, we implement a Retrieval and Re-ranking pipeline. Given a task query, we first employ a retrieval algorithm (i.e., BM25) to perform coarse-grained retrieval from the tutorial library. Then, an off-the-shelf generative re-ranker [7] evaluates the visual similarity between the top-K candidates and the task query alongside textual relevance. This dual-stage process ensures the injection of high quality tutorial into the agent's context, providing essential guidance for zero-shot generalization to unseen websites.

To enhance the model's robustness against potential retrieval noise during inference, we intentionally introduce noise into the training set. Specifically, we randomly pair some tasks with irrelevant or distracting tutorials. Additionally, we withhold tutorials for certain tasks, which requires the agent to complete the task directly using its internal parameter knowledge.

`Weblearner` integrates all three memory levels with the user instruction $\mathbf{I}$ at the current time step t. The integrated multimodal context is fed into an MLLM $F_\theta(\cdot)$ to generate output v_t via auto-regressive decoding:

$$[c_t, r_t, \alpha_t] = v_t \sim F_\theta(\mathbf{M}^t_{kb}, \mathbf{M}^t_{prog}, \mathbf{M}^t_{obs}, \mathbf{I}). \tag{1}$$

Then the output v_t is decomposed into three components: **(1)** a Chain-of-Thought based analysis and summary c_t regarding tutorials and observations; **(2)** a reference r_t indicating which subtask in the tutorial contributes to the current state; **(3)** an executable action command α_t with its description.

We define three core subtasks for the agent at each interaction step:

(1) Memory analysis & summarization: The agent first analyzes the hierarchical memory and summarize content beneficial to current action prediction

and the final answer. While multimodal data in the short-term observational buffer may be overwritten, the agent-generated summaries are preserved in the progress tracker to support subsequent reasoning.
(2) Tutorial-based skill learning: The agent is forced to learn from the multimodal tutorials. Specifically, it is required to identify which tutorial subtasks are pertinent to the current state before determining the next action.
(3) Reasoning integration & action execution: Finally, the agent integrates all reasoning, generates a one-sentence description of the intended action, and produces an executable action command.
Reward modeling. We optimize `weblearner` using the GRPO algorithm [53], which eliminates the need for chain-of-thought supervision. To effectively guide the agent's learning from tutorials and its decision-making, we propose three complementary reward functions: **(1) Format reward ($R^f(\cdot) \in \mathbb{R}$):** Ensures the structural correctness of the output. **(2) Tutorial reference reward ($R^r(\cdot) \in \mathbb{R}$):** Evaluates the accuracy of the generated references to the tutorial subtasks. **(3) Action reward ($R^a(\cdot) \in \mathbb{R}$):** Provides binary feedback on the correctness of the action's type, format, and spatial location.
(1) Format reward. The required output must conform to a strict XML-style format containing these four tags in sequence: *"<analysis></analysis>"*, *"<reference><reference>"*, *"<action_desc></action_desc>"*, *"<action></action>"*. We define a function, $CheckFormats(\cdot)$, which returns *True* if and only if the agent's output strictly adheres to this predefined tag sequence and structure.

$$r^f_{t,i} = R^f(v_{t,i}) = \begin{cases} 1 & \text{if } CheckFormat(v_{t,i}) \\ 0 & \text{else} \end{cases}. \tag{2}$$

(2) Reference reward. We employ a model-based evaluation approach to generate reference reward. Our judge model takes the multimodal tutorial, current observation, state, and the agent's predicted reference as input, and provides a binary feedback signal:

$$r^r_{t,i} = R^r(r_{t,i}, \mathbf{M}_{kb}, \mathbf{M}_{obs}). \tag{3}$$

(3) Action reward. A rule-based evaluation function is used to assess the alignment between the predicted action and ground truth action in terms of action type and arguments:

$$r^a_{t,i} = R^a(a_{t,i}, a^{gt}). \tag{4}$$

Reinforced training. Our total reward combines evaluations of output format correctness, action accuracy, and reference accuracy. It is defined as

$$r_{t,i} = r^f_{t,i} + \lambda^r \cdot r^r_{t,i} + \lambda^a \cdot r^a_{t,i}, \tag{5}$$

where λ^r and λ^a are the weights for the reference and action rewards, respectively. Then, we compute the advantage $A_{t,i}$ via a group-level normalization as

$$A_{t,i} = \frac{r_i - \text{mean}(\{r_1, r_2, ..., r_n\})}{\text{std}(\{r_1, r_2, ..., r_n\})}, \tag{6}$$

and optimize the GRPO training objective via policy gradient descent.

5 Experiments

5.1 Experimental Settings

Benchmarks & Baselines. We evaluate our method on our proposed `WebOne` benchmark, alongside two open source datasets: Multimodal Mind2Web [8] and VideoWebArena [23]. For `WebOne`, we select proprietary models including GPT-5 [49], Claude-sonnet-4-6, and Gemini 2.5 Pro [7], as well as open source LMMs such as Qwen3-VL-8B [3]. We also include specialized GUI baselines like Open-WebVoyager [19] and WorldGUI [81]. For both proprietary and open source LMMs, we implement two configurations: a vanilla setting where the model completes tasks directly, and a RAG setting that utilizes the retrieval and reranking pipeline described in Sec. 4 to match the most relevant tutorials. For Multimodal Mind2Web, we follow previous work and select RAG-GUI [71] as our primary baseline. For VideoWebArena, we apply the same methodology to extract multimodal tutorials from videos as detailed in the appendix. Consequently, we employ the same set of baselines as in `WebOne`. Additionally, we include methods that learn directly from instructional videos as baseline comparisons. We use default settings to ensure a fair evaluation unless otherwise specified. We carefully tune the prompts for all baselines to achieve optimal results, and the full prompt details are provided in the appendix.

Evaluation metrics. Following recent practice [18, 82], we adopt *task success rate* as the evaluation metric for `WebOne`, which is automatically assessed by GPT-4o [1]. We input the task description, the agent model's final answer, and a sequence of screenshots into the large language model and prompt it to determine whether the agent has completed the task. For Multimodal Mind2Web and VideoWebArena, we adopt the default evaluation metrics proposed in their original papers. Specifically, the metrics for Mind2Web include Element Accuracy (Ele.Acc) and Step Success Rate (SSR), while for VideoWebArena, we use the Task Success Rate. The evaluation prompt is provided in the appendix.

Implementation details. Our `WebLearner` is model agnostic. In our main experiments, we employ Qwen3-VL-8B as the base model. The parameters of the visual backbone are frozen, with only the language model being fine-tuned. Our experiments are conducted with a batch size of 32 and a learning rate of 1e-6 on a server equipped with four A800 GPUs, spanning a training period of 24 hours. We set the short-term observational buffer size k to 2, the weights λ^r and λ^a are 0.2 and 0.8, respectively. The maximum number of iterations is set to 15. Other details are provided in the appendix.

5.2 Results on `WebOne`.

As reported in Table 1, our `WebLearner` obtains an overall task success rate of 56.9%, which is 20.1% higher than Qwen3-VL-8B under the same setting. This achieves the state-of-the-art result among open-sourced models and comparable results against proprietary models using vanilla settings. This suggests that introducing external multimodal tutorials is the key to transforming blind

Method	Mode	ESPN	MBTA	arXiv	Uniqlo	HF	Flight
GPT-5 [49]	Vanilla	32.2	46.0	49.5	36.8	47.1	39.5
Claude-sonnet-4-6	Vanilla	33.3	58.0	65.1	39.3	48.7	59.5
Claude-sonnet-4-6	RAG	38.4	65.3	71.4	43.2	50.0	64.2
Gemini-2.5-Pro [7]	Vanilla	44.4	54.0	48.8	44.6	60.0	81.8
Gemini-2.5-Pro [7]	RAG	44.4	70.0	76.7	58.9	80.0	93.2
WorldGUI [81] (w/Gemini)	RAG	44.7	58.0	59.5	55.4	70.6	90.0
OpenWebVovager(OWV) [19]	Vanilla	0.0	15.6	11.3	8.4	20.0	9.4
(w/Qwen3+SFT) [19]	Vanilla	33.3	46.2	43.4	28.9	41.1	49.1
OWV(w/Qwen3+GRPO) [19]	Vanilla	39.5	66.6	45.7	39.6	48.3	48.7
Qwen3-VL-8B [3]	Vanilla	16.7	44.0	37.2	23.2	48.6	63.6
Qwen3-VL-8B [3]	RAG	15.6	52.0	44.2	23.2	39.4	65.9
WebLearner (Ours)	RAG	**33.3**	**74.0**	**65.1**	**42.9**	**68.6**	**75.0**

Method	Mode	Spothero	RT	Book	Yahoo	Ticket	Overall
GPT-5 [49]	Vanilla	31.3	40.0	18.8	35.0	28.7	37.1
Claude-sonnet-4-6	Vanilla	42.6	58.0	34.3	50.0	37.3	48.7
Claude-sonnet-4-6	RAG	39.3	61.2	34.3	63.3	50.6	53.2
Gemini-2.5-Pro [7]	Vanilla	55.7	56.0	33.3	53.3	48.0	53.7
Gemini-2.5-Pro [7]	RAG	57.3	60.0	41.6	68.3	56.0	65.1
WorldGUI [81] (w/Gemini)	RAG	47.5	74.0	45.5	80.7	66.2	63.1
OpenWebVovager(OWV) [19]	Vanilla	13.9	10.8	0.0	11.6	17.4	12.4
(w/Qwen3+SFT) [19]	Vanilla	37.5	45.0	21.8	40.7	51.2	41.5
OWV(w/Qwen3+GRPO) [19]	Vanilla	48.1	61.9	21.8	45.6	45.5	45.8
Qwen3-VL-8B [3]	Vanilla	32.8	44.0	8.3	48.3	29.3	37.8
Qwen3-VL-8B [3]	RAG	36.1	42.0	8.3	33.3	28.0	36.8
WebLearner (Ours)	RAG	**48.8**	**66.0**	**28.0**	**55.0**	**59.5**	**56.9**

Table 1: Comparison against strong baselines on real-world websites. "Overall" denotes the overall performance across all websites. "Vanilla" denotes performing tasks directly without tutorials, while "RAG" indicates the use of retrieved tutorials. Note that "OWV" utilizes in-domain training data. In contrast, for `WebLearner`, all tested websites remain unseen during the training phase.

execution into informed navigation, providing a promising path toward truly generalized web GUI agents.

Proprietary models also obtain significant improvements (over 11%) when using multimodal tutorials. This demonstrates the high quality of our curated multimodal tutorials, which bring additional performance gains even to strong models. Furthermore, it reveals the capability of proprietary models to actively learn from the provided context.

Interestingly, we observe that Qwen3 VL 8B fails to benefit from multimodal tutorials without SFT or RL. Its overall performance actually decreases from 37.8% to 36.8%. We find that Qwen3 VL 8B tends to extract cues solely from the local context and generates actions directly, ignoring the multimodal tutorials provided earlier in the sequence. Furthermore, because the tutorials are related but not identical to the current task, the model must carefully reason about their

Method	Mode	ESPN	MBTA	arXiv	Uniqlo	HF	Flight
WebLearner (Full Model)	RAG	33.3	74.0	65.1	42.9	68.6	75.0
w/o MM Tutorial	Vanilla	28.9	65.3	57.1	42.3	42.4	60.5
w/o Progress Tracker	RAG	34.4	50.0	44.7	34.6	51.6	65.9
w/o Reference Rewards	RAG	34.4	61.9	62.2	23.9	43.3	72.2
w/o Format Rewards	RAG	28.9	52.0	44.2	22.2	47.0	72.7
GRPO → SFT	RAG	21.1	52.0	32.6	42.9	38.2	34.1
MM → Text Tutorial	RAG	26.3	71.4	55.5	38.7	47.8	40.7

Method	Mode	Spothero	RT	Book	Yahoo	Ticket	Overall
WebLearner (Full Model)	RAG	55.7	66.0	28.0	55.0	59.5	56.9
w/o MM Tutorial	Vanilla	31.5	55.3	32.3	40.4	36.4	44.6
w/o Progress Tracker	RAG	43.4	66.0	27.6	43.4	44.1	46.0
w/o Reference Rewards	RAG	48.9	63.4	17.9	50.9	46.0	48.4
w/o Format Rewards	RAG	38.3	60.0	12.5	43.4	43.8	43.0
GRPO → SFT	RAG	45.9	38.0	6.5	26.7	38.7	42.3
MM → Text Tutorial	RAG	37.1	71,9	25.0	31.3	51.1	47.1

Table 2: Ablation study on `WebOne`. "w/o" denotes "without", and "$a \rightarrow b$" indicates replacing a with b. The "w/o tutorial" configuration corresponds to the "Vanilla" setting in our main experiments.

relevance to gain any benefit. Open source models, however, tend to engage in simple mimicry rather than active learning from these tutorials.

We also present baselines using 1K in-domain trajectory data for SFT. Open-WebVoyager achieves a clear improvement of 3.7% compared to Qwen3-VL-8B. However, collecting large-scale in-domain trajectories is expensive. With limited SFT data, models tend to memorize in-domain training samples rather than learning how to reason. Although this improves performance on downstream tasks, these models face obvious bottlenecks when encountering new websites.

We observe that task difficulty varies across different websites. Even within the travel domain, the task success rate on Booking is much lower than on Google Flights. We find this is mainly due to differences in task nature. For instance, booking a flight usually involves a clear intent, while booking a hotel often requires satisfying multiple decision conditions. We note that lower performing websites typically contain many complex user instructions with multiple constraints (e.g., ESPN, arXiv, and Booking). It is interesting to observe that OWV presents a performance of 0.0% on both ESPN and Booking. While OWV identifies some results that partially meet user instructions, it still struggles to find results that fully satisfy all criteria. Baseline models often terminate the task after finding an item that satisfies only one condition without exploring other requirements, which is considered a failure. We conclude that complex user instructions containing multiple decision conditions remain challenging for both our model and the baseline methods.

5.3 Ablation Study

We conduct ablation experiments on the `WebOne` test set to evaluate the contribution of each component in our framework. The results for different websites are shown in Table 2.

The experience memory, which provides multimodal tutorials, is vital for cross-website generalization. Removing this module leads to a performance drop of over 10% in 6 out of 14 websites. On the Ticket website, the success rate decreases significantly from 59.5% to 36.4%. These results show that the agent relies on external knowledge from tutorials to navigate unfamiliar interfaces. The procedure memory is important for maintaining task consistency. When this component is removed, the overall performance drops from 56.9% to 46.0%. Significant decreases (over 10%) are observed in 5 out of 12 websites. For example, on the MBTA website, the success rate falls from 74.0% to 50.0%. This indicates that recording historical states is essential for handling complex, long-horizon tasks.

We also analyze the contribution of different reward types used during training and observe that the reference reward is crucial for task success. This reward encourages the agent to follow the prescribed tutorial steps; without it, the model lacks the necessary supervised signals to effectively learn from multimodal tutorials and align its actions with expert knowledge, leading the overall performance to drop to 48.4%. We observe a significant performance decline when the format reward is removed. Without this constraint, the model's output fails to maintain its original structure. Instead, the model tends to generate non-executable or disorganized responses, which leads to a sharp decrease in the task success rate, particularly on complex websites.

We remove the images from the original tutorials and retain only the textual guidance. We find that the overall performance decreases by 9.8%. This suggests that while the text provides global planning and procedural guidance, the visual information illustrates specific UI transitions. These visual cues help the agent ground itself in tutorial steps, identify target elements, and detect execution failures or termination states. Furthermore, we evaluate our model using Supervised Fine-Tuning (SFT) instead of Reinforced Fine-Tuning (RFT) on the same trajectory training data. The results show that performance drops by 14.6% under the SFT setting. We observe that the SFT-trained agent tends to repeat the same incorrect actions rather than recovering from failures. This aligns with prior findings [34] that RFT is more data-efficient than SFT. With limited data, SFT often leads to overfitting on specific patterns, whereas RFT improves the model's generalization capabilities.

5.4 Results on `VideoWebArena` and `Multimodal Mind2Web`

As shown in Table 3, we observe that the baseline, which learns directly from raw instructional videos, yields limited improvements on `VideoWebArena`. We attribute this to two primary factors: 1) the information in raw videos is highly sparse, making it difficult for the model to extract useful guidance from long

Models	Sources	Classified	Shopping	ShopAdmin	Overall
Gemini-2.5-Pro [7]	Raw Videos	53.3	54.5	46.8	52.6
GPT-5 [49]	None	50.0	51.7	38.3	48.5
Gemini-2.5-Pro [7]	None	55.0	53.7	51.1	53.5
Gemini-2.5-Pro [7]	Tutorials (Ours)	58.3	57.0	57.4	57.4
Qwen3-VL-8B [3]	None	40.0	36.4	27.7	35.6
Qwen3-VL-8B [3]	Tutorials (Ours)	45.0	41.3	34.0	40.8
WebLearner	Tutorials (Ours)	48.3	48.8	46.8	48.3

Table 3: Experiments on VideoWebArena. "Tutorials (Ours)" denotes the tutorials constructed using our automated pipeline from raw video data.

sequences; 2) video input significantly extends the sequence length from approximately 6k to 100k tokens, which may lead to performance degradation in long context processing.

In contrast, our multimodal tutorials provide significant performance gains. These tutorials serve as an information-dense external knowledge source that offers procedural guidance, effectively outperforming sparse raw videos. In addition, the automated tutorial construction pipeline developed for `WebOne` can be seamlessly adapted to other data sources, demonstrating the potential of our approach for broader real-world applications.

As shown in Table 4, we further evaluate `WebLearner`'s performance on `Multimodal Mind2Web` dataset. We observe that `WebLearner` achieves competitive improvements over other strong baselines. In terms of the Step Success Rate (SSR) metric, our model consistently outperforms all baselines across all three subsets. Notably, WebLearner is tested directly on this dataset without using its training data. Due to the differences in experimental settings, our tutorial repository does not cover all the tasks required by Mind2Web. For samples where no relevant tutorial is found, our generative re-ranker outputs "None". This indicates the absence of a matching tutorial and requires the agent to complete the task using its internal knowledge. Our method is capable of handling such cases because we explicitly incorporated scenarios with irrelevant or missing tutorials during the training phase.

5.5 Case Study

As shown in Figure 3, we showcase a randomly-selected example in which Qwen3-VL-8B and our `WebLearner` perform an advanced search task on the arXiv website. Due to space limitations, we have omitted the thinking process of our model. Our method successfully completed the task using the minimum number of clicks (five times). However, despite being provided with the tutorial, Qwen3-VL-8B failed to effectively extract helpful information from it, such as global planning. Qwen3-VL-8B attempted several buttons on the arXiv category page without success. The model then directly used the general search bar to look up the

Subsets	Cross Task		Cross Website		Cross Domain	
Models	Ele.Acc	SSR	Ele.Acc	SSR	Ele.Acc	SSR
GPT-4V+OmniParse [42]	42.4	39.4	41.0	36.5	45.5	42.0
Claude Com. Use [2]	62.7	53.5	59.5	47.7	64.5	56.4
ShowUI [30]	39.9	37.2	41.6	35.1	39.4	35.2
AgentTrek [72]	45.5	40.9	40.8	35.1	48.6	42.1
Magma [74]	54.8	43.4	57.2	45.4	55.7	47.3
RAG-GUI [71]	63.6	51.5	60.7	46.1	60.3	47.8
WebLearner	63.4	53.4	61.4	51.9	62.0	54.0

Table 4: Experiments on Multimodal Mind2Web. Note that our `WebLearner` is trained on `WebOne` and does not utilize training data from Mind2Web for training.

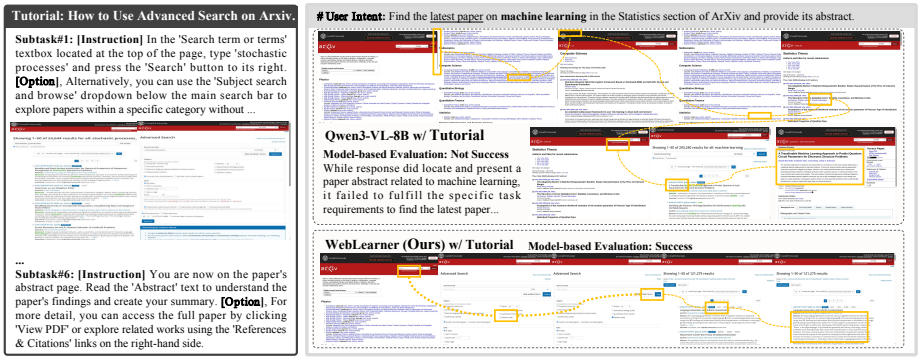

Fig. 3: A randomly-selected case where the agents are prompted to perform an advanced search on arXiv website.

keyword "machine learning". After obtaining the search results, it terminated the task and returned the first result. However, this search result only partially met the requirements of the user instruction and was therefore judged as not successful.

6 Conclusion

In this paper, we address the critical challenge of generalization in web GUI agents navigating dynamic and unfamiliar environments. We introduce a hierarchical memory framework that utilizes multimodal tutorials to enhance test time generalizability, enabling agents to acquire novel skills and adapt to unseen interfaces without further training. Our contributions are threefold. First, we present `WebOne`, a comprehensive dataset comprising 1342 online tasks and 970 high-quality multimodal tutorials across 60+ real-world websites. Second,

we propose `WebLearner`, a reinforcement learning based agent that effectively synthesizes information from a multimodal knowledge base and a dynamic task tracker to execute long-horizon tasks. Finally, empirical evaluations demonstrate that `WebLearner` significantly outperforms strong baseline models.

Acknowledgments.

This work was supported by the National Science and Technology Major Project (2023ZD0121403), and NSFC No.62306162.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Anthropic, S.: Model card addendum: Claude 3.5 haiku and upgraded claude 3.5 sonnet. URL https://api. semanticscholar. org/CorpusID **273639283**, 24 (2024)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report (2025)
4. Chen, G., Zhou, X., Shao, R., Lyu, Y., Zhou, K., Wang, S., Li, W., Li, Y., Qi, Z., Nie, L.: Less is more: Empowering gui agent with context-aware simplification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5901–5911 (2025)
5. Chen, W., Cui, J., Hu, J., Qin, Y., Fang, J., Zhao, Y., Wang, C., Liu, J., Chen, G., Huo, Y., et al.: Guicourse: From general vision language model to versatile gui agent. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 21936–21959 (2025)
6. Cheng, K., Sun, Q., Chu, Y., Xu, F., Li, Y., Zhang, J., Wu, Z.: Seeclick: Harnessing gui grounding for advanced visual gui agents. arXiv preprint arXiv:2401.10935 (2024)
7. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
8. Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., Su, Y.: Mind2web: Towards a generalist agent for the web. Advances in Neural Information Processing Systems **36**, 28091–28114 (2023)
9. Fan, G., Niu, C., Lyu, C., Wu, F., Chen, G.: Core: Reducing ui exposure in mobile agents via collaboration between cloud and local llms. arXiv preprint arXiv:2510.15455 (2025)
10. Fan, Y., Zhang, K., Zhou, H., Zuo, Y., Chen, Y., Fu, Y., Long, X., Zhu, X., Jiang, C., Zhang, Y., et al.: Ssrl: Self-search reinforcement learning. arXiv preprint arXiv:2508.10874 (2025)

11. Gao, D., Ji, L., Bai, Z., Ouyang, M., Li, P., Mao, D., Wu, Q., Zhang, W., Wang, P., Guo, X., et al.: Assistgui: Task-oriented pc graphical user interface automation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13289–13298 (2024)
12. Gao, L., Zhang, L., Xu, M.: Uishift: Enhancing vlm-based gui agents through self-supervised reinforcement learning. arXiv preprint arXiv:2505.12493 (2025)
13. Gao, Y., Ye, J., Wang, J., Sang, J.: Websynthesis: World-model-guided mcts for efficient webui-trajectory synthesis. arXiv preprint arXiv:2507.04370 (2025)
14. Garkot, S., Shamrai, M., Synytsia, I., Hirna, M.: Guirilla: A scalable framework for automated desktop ui exploration. arXiv preprint arXiv:2510.16051 (2025)
15. Ge, Z., Li, J., Pang, X., Gao, M., Pan, K., Lin, W., Fei, H., Zhang, W., Tang, S., Zhuang, Y.: Iris: Breaking gui complexity with adaptive focus and self-refining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24559–24568 (2025)
16. Guan, Z., Li, J.C.L., Hou, Z., Zhang, P., Xu, D., Zhao, Y., Wu, M., Chen, J., Nguyen, T.T., Xian, P., et al.: Kg-rag: Enhancing gui agent decision-making via knowledge graph-driven retrieval-augmented generation. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 5396–5405 (2025)
17. He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., Yu, D.: Webvoyager: Building an end-to-end web agent with large multimodal models. arXiv preprint arXiv:2401.13919 (2024)
18. He, H., Yao, W., Ma, K., Yu, W., Zhang, H., Fang, T., Lan, Z., Yu, D.: Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. arXiv preprint arXiv:2410.19609 (2024)
19. He, H., Yao, W., Ma, K., Yu, W., Zhang, H., Fang, T., Lan, Z., Yu, D.: Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. In: Proceedings of the 63rd ACL. pp. 27545–27564 (2025)
20. Heiskanen, A.: Robotic process automation in automated gui testing of web applications (2021)
21. Hou, X., Zhao, Y., Wang, S., Wang, H.: Model context protocol (mcp): Landscape, security threats, and future research directions. ACM Transactions on Software Engineering and Methodology (2025)
22. Huang, Z., Cheng, Z., Pan, J., Hou, Z., Zhan, M.: Spiritsight agent: Advanced gui agent with one look. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29490–29500 (2025)
23. Jang, L., Li, Y., Zhao, D., Ding, C., Lin, J., Liang, P.P., Bonatti, R., Koishida, K.: Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. arXiv preprint arXiv:2410.19100 (2024)
24. Jia, H., Liao, J., Zhang, X., Xu, H., Xie, T., Jiang, C., Yan, M., Liu, S., Ye, W., Huang, F.: Osworld-mcp: Benchmarking mcp tool invocation in computer-use agents. arXiv preprint arXiv:2510.24563 (2025)
25. Kil, J., Song, C.H., Zheng, B., Deng, X., Su, Y., Chao, W.L.: Dual-view visual contextualization for web navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14445–14454 (2024)
26. Koh, J.Y., Lo, R., Jang, L., Duvvur, V., Lim, M.C., Huang, P.Y., Neubig, G., Zhou, S., Salakhutdinov, R., Fried, D.: Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. arXiv preprint arXiv:2401.13649 (2024)

27. Lai, H., Liu, X., Zhao, Y., Xu, H., Zhang, H., Jing, B., Ren, Y., Yao, S., Dong, Y., Tang, J.: Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. arXiv preprint arXiv:2508.14040 (2025)
28. Li, Y., Hultquist, H., Wagle, J., Koishida, K.: Instruction agent: Enhancing agent with expert demonstration. arXiv preprint arXiv:2509.07098 (2025)
29. Lin, K.Q., Li, L., Gao, D., Wu, Q., Yan, M., Yang, Z., Wang, L., Shou, M.Z.: Videogui: A benchmark for gui automation from instructional videos. arXiv preprint arXiv:2406.10227 **4** (2024)
30. Lin, K.Q., Li, L., Gao, D., Yang, Z., Wu, S., Bai, Z., Lei, S.W., Wang, L., Shou, M.Z.: Showui: One vision-language-action model for gui visual agent. In: Proceedings of the CVPR. pp. 19498–19508 (2025)
31. Liu, G., Zhao, P., Liu, L., Chen, Z., Chai, Y., Ren, S., Wang, H., He, S., Meng, W.: Learnact: Few-shot mobile gui agent with a unified demonstration benchmark. arXiv preprint arXiv:2504.13805 (2025)
32. Liu, H., Zhang, X., Xu, H., Wanyan, Y., Wang, J., Yan, M., Zhang, J., Yuan, C., Xu, C., Hu, W., et al.: Pc-agent: A hierarchical multi-agent collaboration framework for complex task automation on pc. arXiv preprint arXiv:2502.14282 (2025)
33. Liu, T., Wang, C., Li, R., Yu, Y., He, X., Song, B.: Gui-rise: Structured reasoning and history summarization for gui navigation. arXiv preprint arXiv:2510.27210 (2025)
34. Liu, Z., Sun, Z., Zang, Y., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual-rft: Visual reinforcement fine-tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2034–2044 (2025)
35. Long, X., Ma, Z., Hua, E., Zhang, K., Qi, B., Zhou, B.: Retrieval-augmented visual question answering via built-in autoregressive search engines. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 24723–24731 (2025)
36. Long, X., Zeng, J., Meng, F., Ma, Z., Zhang, K., Zhou, B., Zhou, J.: Generative multi-modal knowledge retrieval with large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 18733–18741 (2024)
37. Loo, G., Liu, C., Yin, Q., Chen, X., Chen, J., Zhang, J., Tian, Y.: Mobile-rag: Enhancing mobile agent with retrieval-augmented generation. arXiv preprint arXiv:2509.03891 (2025)
38. Lu, D., Xu, Y., Wang, J., Wu, H., Wang, X., Wang, Z., Yang, J., Su, H., Chen, J., Chen, J., et al.: Videoagenttrek: Computer use pretraining from unlabeled videos. arXiv preprint arXiv:2510.19488 (2025)
39. Lu, F., Zhong, Z., Liu, S., Fu, C.W., Jia, J.: Arpo: End-to-end policy optimization for gui agents with experience replay. arXiv preprint arXiv:2505.16282 (2025)
40. Lu, Q., Ma, Z., Zhong, S., Wang, J., Yu, D., Ng, M.K., Luo, P.: Swirl: A staged workflow for interleaved reinforcement learning in mobile gui control. arXiv preprint arXiv:2508.20018 (2025)
41. Lu, Q., Shao, W., Liu, Z., Du, L., Meng, F., Li, B., Chen, B., Huang, S., Zhang, K., Luo, P.: Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22404–22414 (2025)
42. Lu, Y., Yang, J., Shen, Y., Awadallah, A.: Omniparser for pure vision based gui agent. arXiv preprint arXiv:2408.00203 (2024)
43. Lu, Y., Huang, J., Liu, H., Gesi, J., Han, Y., Fu, S., Zheng, T., Wang, D.: Webserv: A browser-server environment for efficient training of reinforcement learning-based web agents at scale. arXiv preprint arXiv:2510.16252 (2025)

44. Lu, Z., Chai, Y., Guo, Y., Yin, X., Liu, L., Wang, H., Xiao, H., Ren, S., Xiong, G., Li, H.: Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. arXiv preprint arXiv:2503.21620 (2025)
45. Lu, Z., Ye, J., Tang, F., Shen, Y., Xu, H., Zheng, Z., Lu, W., Yan, M., Huang, F., Xiao, J., et al.: Ui-s1: Advancing gui automation via semi-online reinforcement learning. arXiv preprint arXiv:2509.11543 (2025)
46. Luo, R., Wang, L., He, W., Chen, L., Li, J., Xia, X.: Gui-r1: A generalist r1-style vision-language action model for gui agents. arXiv preprint arXiv:2504.10458 (2025)
47. Memon, A., Banerjee, I., Hashmi, N., Nagarajan, A.: Dart: a framework for regression testing "nightly/daily builds" of gui applications. In: International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings. pp. 410–419. IEEE (2003)
48. Nguyen, D., Chen, J., Wang, Y., Wu, G., Park, N., Hu, Z., Lyu, H., Wu, J., Aponte, R., Xia, Y., et al.: Gui agents: A survey. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 22522–22538 (2025)
49. OpenAI: Introducing GPT-5 (8 2025), accessed: 2025-08-07
50. Pan, Y., Kong, D., Zhou, S., Cui, C., Leng, Y., Jiang, B., Liu, H., Shang, Y., Zhou, S., Wu, T., et al.: Webcanvas: Benchmarking web agents in online environments. arXiv preprint arXiv:2406.12373 (2024)
51. Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al.: Ui-tars: Pioneering automated gui interaction with native agents. arXiv preprint arXiv:2501.12326 (2025)
52. Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W., Li, W., Campbell-Ajala, F., et al.: Androidworld: A dynamic benchmarking environment for autonomous agents. arXiv preprint arXiv:2405.14573 (2024)
53. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
54. Shi, Y., Yu, W., Li, Z., Wang, Y., Zhang, H., Liu, N., Mi, H., Yu, D.: Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment. arXiv preprint arXiv:2507.05720 (2025)
55. Song, Y., Xu, F., Zhou, S., Neubig, G.: Beyond browsing: Api-based web agents (2025)
56. Sun, Q., Li, M., Liu, Z., Xie, Z., Xu, F., Yin, Z., Cheng, K., Li, Z., Ding, Z., Liu, Q., et al.: Os-sentinel: Towards safety-enhanced mobile gui agents via hybrid validation in realistic workflows. arXiv preprint arXiv:2510.24411 (2025)
57. Sun, Y., Zhao, S., Yu, T., Wen, H., Va, S., Xu, M., Li, Y., Zhang, C.: Gui-xplore: Empowering generalizable gui agents with one exploration. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 19477–19486 (2025)
58. Sun, Z., Cao, Y., Liang, J., Sun, Q., Liu, Z., Zhang, Z., Zang, Y., Dong, X., Chen, K., Lin, D., et al.: Coda: Coordinating the cerebrum and cerebellum for a dual-brain computer use agent with decoupled reinforcement learning. arXiv preprint arXiv:2508.20096 (2025)
59. Sun, Z., Liu, Z., Zang, Y., Cao, Y., Dong, X., Wu, T., Lin, D., Wang, J.: Seagent: Self-evolving computer use agent with autonomous learning from experience. arXiv preprint arXiv:2508.04700 (2025)
60. Tao, Z., Shen, H., Li, B., Yin, W., Wu, J., Li, K., Zhang, Z., Yin, H., Ye, R., Zhang, L., et al.: Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. arXiv preprint arXiv:2510.24697 (2025)

61. Wang, H., Zou, H., Song, H., Feng, J., Fang, J., Lu, J., Liu, L., Luo, Q., Liang, S., Huang, S., et al.: Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. arXiv preprint arXiv:2509.02544 (2025)
62. Wang, J., Zhou, J., Zhang, W., Liu, W., Zhang, Z., Lou, X., Zhang, W., Deng, H., Wang, J.: Colorbrowseragent: Complex long-horizon browser agent with adaptive knowledge evolution (2026)
63. Wang, J., Xu, H., Zhang, X., Yan, M., Zhang, J., Huang, F., Sang, J.: Mobile-agent-v: Learning mobile device operation through video-guided multi-agent collaboration. arXiv preprint arXiv:2502.17110 (2025)
64. Wang, Z., Chen, W., Yang, L., Zhou, S., Zhao, S., Zhan, H., Jin, J., Li, L., Shao, Z., Bu, J.: Mp-gui: Modality perception with mllms for gui understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29711–29721 (2025)
65. Wanyan, Y., Zhang, X., Xu, H., Liu, H., Wang, J., Ye, J., Kou, Y., Yan, M., Huang, F., Yang, X., et al.: Look before you leap: A gui-critic-r1 model for pre-operative error diagnosis in gui automation. arXiv preprint arXiv:2506.04614 (2025)
66. Wei, J., Liu, J., Liu, L., Hu, M., Ning, J., Li, M., Yin, W., He, J., Liang, X., Feng, C., et al.: Learning, reasoning, refinement: A framework for kahneman's dual-system intelligence in gui agents. arXiv preprint arXiv:2506.17913 (2025)
67. Wu, J., Yin, W., Jiang, Y., Wang, Z., Xi, Z., Fang, R., Zhang, L., He, Y., Zhou, D., Xie, P., et al.: Webwalker: Benchmarking llms in web traversal. arXiv preprint arXiv:2501.07572 (2025)
68. Wu, W., Zhou, K., Yuan, R., Yu, V., Wang, S., Hu, Z., Huang, B.: Auto-scaling continuous memory for gui agent. arXiv preprint arXiv:2510.09038 (2025)
69. Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T.J., Cheng, Z., Shin, D., Lei, F., et al.: Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. Advances in Neural Information Processing Systems **37**, 52040–52094 (2024)
70. Xie, Y., Zhang, X., Shan, Y., Zhu, H., Tang, R., Wei, R., Song, M., Wan, Y., Song, J.: Spatialqa: A benchmark for evaluating spatial logical reasoning in vision-language models. arXiv preprint arXiv:2602.20901 (2026)
71. Xu, R., Ma, K., Yu, W., Zhang, H., Ho, J.C., Yang, C., Yu, D.: Retrieval-augmented gui agents with generative guidelines. arXiv preprint arXiv:2509.24183 (2025)
72. Xu, Y., Lu, D., Shen, Z., Wang, J., Wang, Z., Mao, Y., Xiong, C., Yu, T.: Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. arXiv preprint arXiv:2412.09605 (2024)
73. Xue, T., Peng, C., Huang, M., Guo, L., Han, T., Wang, H., Wang, J., Zhang, X., Yang, X., Zhao, D., et al.: Evocua: Evolving computer use agents via learning from scalable synthetic experience. arXiv preprint arXiv:2601.15876 (2026)
74. Yang, J., Tan, R., Wu, Q., Zheng, R., Peng, B., Liang, Y., Gu, Y., Cai, M., Ye, S., Jang, J., et al.: Magma: A foundation model for multimodal ai agents. In: Proceedings of the computer vision and pattern recognition conference. pp. 14203–14214 (2025)
75. Yao, S., Chen, H., Yang, J., Narasimhan, K.: Webshop: Towards scalable real-world web interaction with grounded language agents. Advances in Neural Information Processing Systems **35**, 20744–20757 (2022)
76. Ye, R., Zhang, Z., Li, K., Yin, H., Tao, Z., Zhao, Y., Su, L., Zhang, L., Qiao, Z., Wang, X., et al.: Agentfold: Long-horizon web agents with proactive context management. arXiv preprint arXiv:2510.24699 (2025)

77. Yi, B., Hu, X., Chen, Y., Zhang, S., Yang, H., Wu, F., Wu, F.: Ecoagent: An efficient edge-cloud collaborative multi-agent framework for mobile automation. arXiv preprint arXiv:2505.05440 (2025)
78. Zhang, B., Shang, Z., Gao, Z., Zhang, W., Xie, R., Ma, X., Yuan, T., Wu, X., Zhu, S.C., Li, Q.: Tongui: Building generalized gui agents by learning from multimodal web tutorials. arXiv preprint arXiv:2504.12679 (2025)
79. Zhang, K., Zuo, Y., He, B., Sun, Y., Liu, R., Jiang, C., Fan, Y., Tian, K., Jia, G., Li, P., et al.: A survey of reinforcement learning for large reasoning models. arXiv preprint arXiv:2509.08827 (2025)
80. Zhang, S., Fu, P., Zhang, R., Yang, J., Du, A., Xi, X., Wang, S., Huang, Y., Qin, B., Luo, Z., et al.: Hyperclick: Advancing reliable gui grounding via uncertainty calibration. arXiv preprint arXiv:2510.27266 (2025)
81. Zhao, H.H., Yang, K., Yu, W., Gao, D., Shou, M.Z.: Worldgui: An interactive benchmark for desktop gui automation from any starting point. arXiv preprint arXiv:2502.08047 (2025)
82. Zhou, S., Xu, F.F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al.: Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854 (2023)
83. Zuo, Y., Zhang, K., Sheng, L., Qu, S., Cui, G., Zhu, X., Li, H., Long, X., Hua, E., Qi, B., et al.: Ttrl: Test-time reinforcement learning. Advances in Neural Information Processing Systems **38**, 131459–131483 (2026)