GUI-ReWalk: Massive Data Generation for GUI Agent via Stochastic Exploration and Intent-Aware Reasoning

Taoran Lu[1,*,†], **Minghao Liu**[1,2,*,‡], **Musen Lin**[1,*], **Lichen Yuan**[1,*], **Yuhao Chao**[1], **Yiwei Liu**[1], **Haonan Xu**[1], **Rui Pan**[1], **Lei Yang**[1], **Yu Miao**[1], **Zhaojian Li**[1,†]
[1]ByteDance
[2]University of Chinese Academy of Sciences
{lutaoran, lizhaojian.joeli}@bytedance.com

Abstract

Graphical User Interface (GUI) Agents, powered by large language and vision-language models, hold promise for enabling end-to-end automation in digital environments. However, their progress is fundamentally constrained by the scarcity of scalable, high-quality trajectory data. Existing data collection strategies either rely on costly and inconsistent manual annotations or on synthetic generation methods that trade off between diversity and meaningful task coverage. To bridge this gap, we present **GUI-ReWalk**: a reasoning-enhanced, multi-stage framework for synthesizing realistic and diverse GUI trajectories. GUI-ReWalk begins with a stochastic exploration phase that emulates human trial-and-error behaviors, and progressively transitions into a reasoning-guided phase where inferred goals drive coherent and purposeful interactions. Moreover, it supports multi-stride task generation, enabling the construction of long-horizon workflows across multiple applications. By combining randomness for diversity with goal-aware reasoning for structure, GUI-ReWalk produces data that better reflects the intent-aware, adaptive nature of human-computer interaction. We further train Qwen2.5-VL-7B on the GUI-ReWalk dataset and evaluate it across multiple benchmarks, including Screenspot-Pro, OSWorld/OSWorld-G, UI-Vision, AndroidControl, and GUI-Odyssey. Results demonstrate that GUI-ReWalk enables superior coverage of diverse interaction flows, higher trajectory entropy, and more realistic user intent. These findings establish GUI-ReWalk as a scalable and data-efficient framework for advancing GUI agent research and enabling robust real-world automation.

1 Introduction

The emergence of Vision-Language Models (VLMs) has significantly advanced the capabilities of autonomous agents in perceiving, reasoning, and acting within complex environments [Zhang *et al.*, 2025b]. A promising and increasingly popular research direction is that of GUI Agents, where large language model (LLM)-based agents interact with Graphical user interfaces (GUIs) to accomplish real-world tasks. By bridging visual perception, semantic understanding, and action planning, GUI Agents are poised to unlock a new era of end-to-end automation, transforming how intelligent systems interact with the digital world across domains ranging from productivity to everyday services.

However, the development of GUI Agents is currently constrained by the availability of high-quality training data. Existing GUI agent trajectories are primarily obtained through manual annotation or synthetic generation. Manual collection involves labeling complete action trajectories and defining high-level tasks, a process that is not only time-consuming and labor-intensive, but also susceptible to inconsistencies in quality and style due to varying annotator expertise. On the other hand, synthetic data generation is typically driven by either predefined task goals or random environment interaction. Task-driven approaches offer clear and structured objectives, but suffer from limited scalability and diversity. In contrast, interaction-based methods promote trajectory diversity, yet often lead to overly divergent behaviors that fail to converge on meaningful task outcomes.

Unlike traditional text or image data, GUI trajectories embody rich patterns of human interaction with graphical interfaces. They are not simple Markovian sequences, but rather unfolding narratives shaped by both intention and exploration. Human behavior in GUI environments typically unfolds through the following progressive stages:

- Exploration and boundary probing: When first encountering an unfamiliar application or interface, users often engage in seemingly random tapping, swiping, and navigating actions to test affordances and interface boundaries;
- Goal formulation and pursuit: As users develop clearer objectives, their actions become more directed and intentional, focusing on accomplishing specific tasks through iterative interactions;
- Cross-application coordination: To fulfill more complex goals, users frequently orchestrate multiple apps in tandem;

[†]Corresponding author.
[‡]Work done at ByteDance.

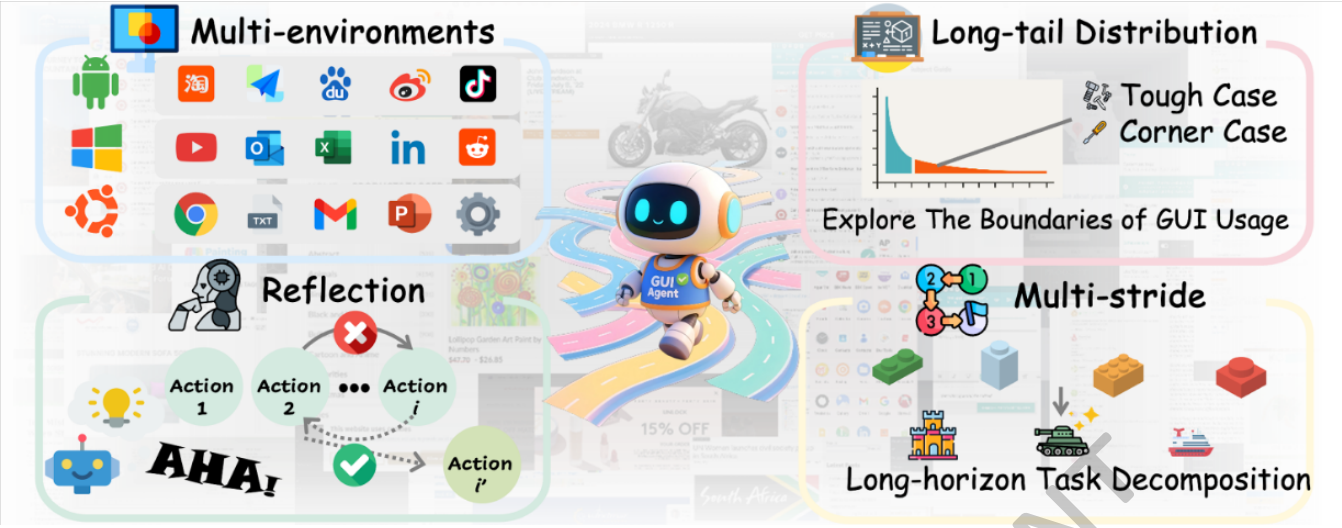

Figure 1: Overview of GUI-ReWalk Characteristics: Multi-platform Coverage, Long-tail Patterns, Reflective Learning, and Multi-stride Workflows.

- Self-correction and backtracking: Users identify missteps or unreachable states and adapt by revising their strategies, undoing actions, or restarting from known checkpoints.

These patterns reveal that GUI trajectories are neither purely random nor rigidly deterministic—they embody a delicate balance between "chaos" and "order," being structured, goal-driven, and highly adaptive.

To address the limitations of existing data acquisition approaches and better capture the nuanced characteristics of human GUI behavior, we propose **G**raphical **U**ser **I**nterface **Re**asoning and random **Walk** (**GUI-ReWalk**)—a multi-stage framework that integrates stochastic exploration with goal-directed reasoning to synthesize diverse and realistic GUI trajectories. Inspired by how humans explore unfamiliar interfaces, GUI-ReWalk begins with a random walk phase, simulating natural trial-and-error behaviors akin to an uninformed policy over a Markov chain, where each state transition depends only on the current state and available actions. As the trajectory unfolds, a large language model (LLM) acts as a reasoning agent that interprets the partially observed sequence and infers high-level goals, transitioning the framework into a reasoning-guided phase. This phase resembles a policy update in a hierarchical Markov Decision Process (hMDP), where action generation is conditioned not only on the current GUI state but also on the inferred intent—mirroring how users refine their behavior upon gaining contextual understanding. In addition, GUI-ReWalk supports multi-stride task generation, where each stride represents a subtask composed of several low-level actions, sequentially coordinated to complete complex goals across multiple interfaces or applications. By unifying the randomness of Markov chains with the intent-aware adaptability of hMDPs, GUI-ReWalk produces synthetic interaction data that captures both the long-tail diversity and the structured, goal-driven nature of real-world human-computer interaction.

In our experiments, we developed GUI-ReWalk-7B, built on Qwen2.5-VL-7B, and trained it on synthetic trajectory data generated within a controlled GUI environment. We evaluated its grounding and navigation capabilities across multiple benchmarks, including Screenspot-Pro, OSWorld-G, and UI-Vision for grounding, and AndroidControl and GUI-Odyssey for navigation. Results demonstrate that GUI-ReWalk, leveraging systematic trajectory generation and task-aware supervision, achieves superior coverage of diverse interaction flows, higher trajectory entropy, and realistic user intent, as validated by human evaluations. These findings establish GUI-ReWalk as a highly effective, scalable, and data-efficient solution for advancing human-computer interaction in diverse GUI environments.

In summary, our work makes the following key contributions:

- **Human-like modeling of GUI interaction:** We formalize GUI trajectories as a hierarchical Markov Decision Process, where each stride combines subgoal abstraction with stride-based reasoning to capture both exploratory and goal-directed behaviors.

- **The GUI-ReWalk framework:** We introduce a multi-stage framework integrating random exploration, task-guided completion, and cross-application task initiation, enhanced by retrospective LLM-based annotation and error-recovery mechanisms that mirror real human interaction patterns.

- **Dataset analysis and model evaluation:** We provide an in-depth analysis of the GUI-ReWalk dataset and demonstrate its effectiveness by training GUI-ReWalk-7B, which achieves substantial improvements across grounding and navigation benchmarks.

2 Related Works

2.1 Evolution of GUI Agents

GUI agents have progressively evolved from rule-based systems to data-driven, end-to-end models. Early approaches—including RPA tools [Dobrica, 2022], DART [Memon *et al.*, 2003] relied on predefined heuristics and API invocations to mimic user actions, but exhibited limited flexibility and poor generalization in dynamic or unfamiliar environments. The emergence of modular agent frameworks—integrating foundation models (e.g., GPT-4o [OpenAI, 2024]), memory systems (e.g., Cradle [Tan *et al.*, 2024]), grounding components (e.g., MM-Navigator [Yan *et al.*, 2023]), and tool-use mechanisms (e.g., AutoGPT [Yang *et al.*, 2023])—enabled more adaptive and multi-step interactions. However, these systems often remained constrained by handcrafted workflows, prompt engineering, and brittle module coordination [Xia *et al.*, 2024].

More recently, native agent architectures such as Claude Computer Use [Anthropic, 2024], Aguvis [Xu *et al.*, 2025], OS-Atlas [Wu *et al.*, 2024], and UI-TARS [Qin *et al.*, 2025] have unified perception, reasoning, memory, and action within end-to-end, vision-centric models. These agents operate directly on raw screenshots without relying on structured UI representations (e.g., accessibility trees or HTML), and are trained on large-scale GUI interaction data, achieving improved generalization across platforms such as web, mobile, and desktop. Building on this foundation, recent work has further enhanced native agents through targeted training strategies—including reinforcement learning, supervised fine-tuning, and curriculum learning—along with dedicated datasets for grounded interaction [Yang *et al.*, 2025], complex task reasoning [Tang *et al.*, 2025], and reflective decision-making [Wu *et al.*, 2025].

2.2 GUI Benchmarks and Environments

Benchmark environments play a central role in the development of GUI agents by defining interaction modalities, task formats, and evaluation protocols. Early benchmarks such as MiniWob++ [Liu *et al.*, 2018] and WoB [Shi *et al.*, 2017] provided synthetic but controlled environments—MiniWob++ emphasized UI layout and instruction diversity, while WoB enabled reproducible task execution on real webpages. Subsequent benchmarks moved toward greater realism and task complexity. WebShop [Yao *et al.*, 2022] introduced compositional shopping tasks requiring semantic reasoning and goal-driven navigation, and Mind2Web [Deng *et al.*, 2023] scaled to 2,000 open-ended tasks across 137 websites with fine-grained step annotations. WebArena [Zhou *et al.*, 2024] simulated multimodal websites across diverse domains (e.g., e-commerce, social media), while WebLINX [Lù *et al.*, 2024] extended to long-horizon, multi-turn workflows using retrieval-augmented prompting and expert demonstrations.

Beyond the browser, benchmarks such as OSWorld [Xie *et al.*, 2024] and WindowsAgentArena [Bonatti *et al.*, 2024] enabled agents to interact with full desktop operating systems, supporting complex workflows like file management and multi-application coordination. On mobile platforms, AndroidWorld [Rawles *et al.*, 2025] and GUI-Odyssey [Lu *et al.*, 2024] enabled fine-grained UI interactions across and within apps. Finally, modality-rich and cross-platform benchmarks have emerged to support generalist agents: GUI-World [Chen *et al.*, 2025] captured video-based GUI behavior grounded in real-world demonstrations, while AgentSynth [Xie *et al.*, 2025a] introduced a modular benchmark that generates long-horizon desktop tasks from atomic subtasks via LLMs, facilitating structured evaluation of planning, perception, and robustness.

2.3 Data Collection and Synthesis for GUI Agents

Training GUI agents depends on large-scale, diverse task trajectories. Early datasets such as WebShop [Yao *et al.*, 2022], Mind2Web [Deng *et al.*, 2023], and AndroidControl [Li *et al.*, 2024] were constructed through human demonstrations to ensure task fidelity and realism. GUI-Odyssey [Lu *et al.*, 2024] further contributed 7,700 mobile interaction episodes spanning both within-app and cross-app workflows. However, the scalability of these human-annotated datasets is hindered by high collection costs.

To overcome this limitation, recent efforts have explored automated data generation techniques. OS-Genesis [Sun *et al.*, 2025] extracts high-quality task trajectories via agent-driven exploration guided by learned reward models. WebSynthesis [Gao *et al.*, 2025] performs world-model-guided search over simulated web interfaces to synthesize interaction traces. GUI-World [Chen *et al.*, 2025] generates video-based interaction data from curated app screenshots, while TongUI [Zhang *et al.*, 2025a] mines web tutorials and converts them into over 143K multimodal, executable task trajectories grounded in realistic application scenarios.

3 Methods

3.1 Overview

The GUI-ReWalk framework is designed to replicate the iterative, exploratory, and goal-oriented behaviors characteristic of human interactions with GUIs. To this end, we formalize the framework as **hierarchical Markov Decision Process**, consisting of multiple sequentially executed steps. Each stride consists of three distinct phases: a random walk phase, a task-guided completion phase, and a task initiation phase in cross-application. The GUI-ReWalk framework integrates both random exploratory actions and logical reasoning processes, thereby enhancing the diversity and length of interaction trajectories. Between these phases, we introduce retrospective annotation, which uses large language models (LLMs) to perform backward annotation and summarization of trajectories, enabling full automation of the process. In both the task-guided completion phase and the task initiation phase, we further incorporate a error task recovery scheme that generates a new goal whenever the LLM becomes blocked in the current environment, drawing upon the previously failed objective and the current state to ensure continuity and robustness in task execution.

Through this mechanism, GUI-ReWalk effectively generates multi-stride trajectories that closely mirror human multi-application workflows, preserving logical task coherence

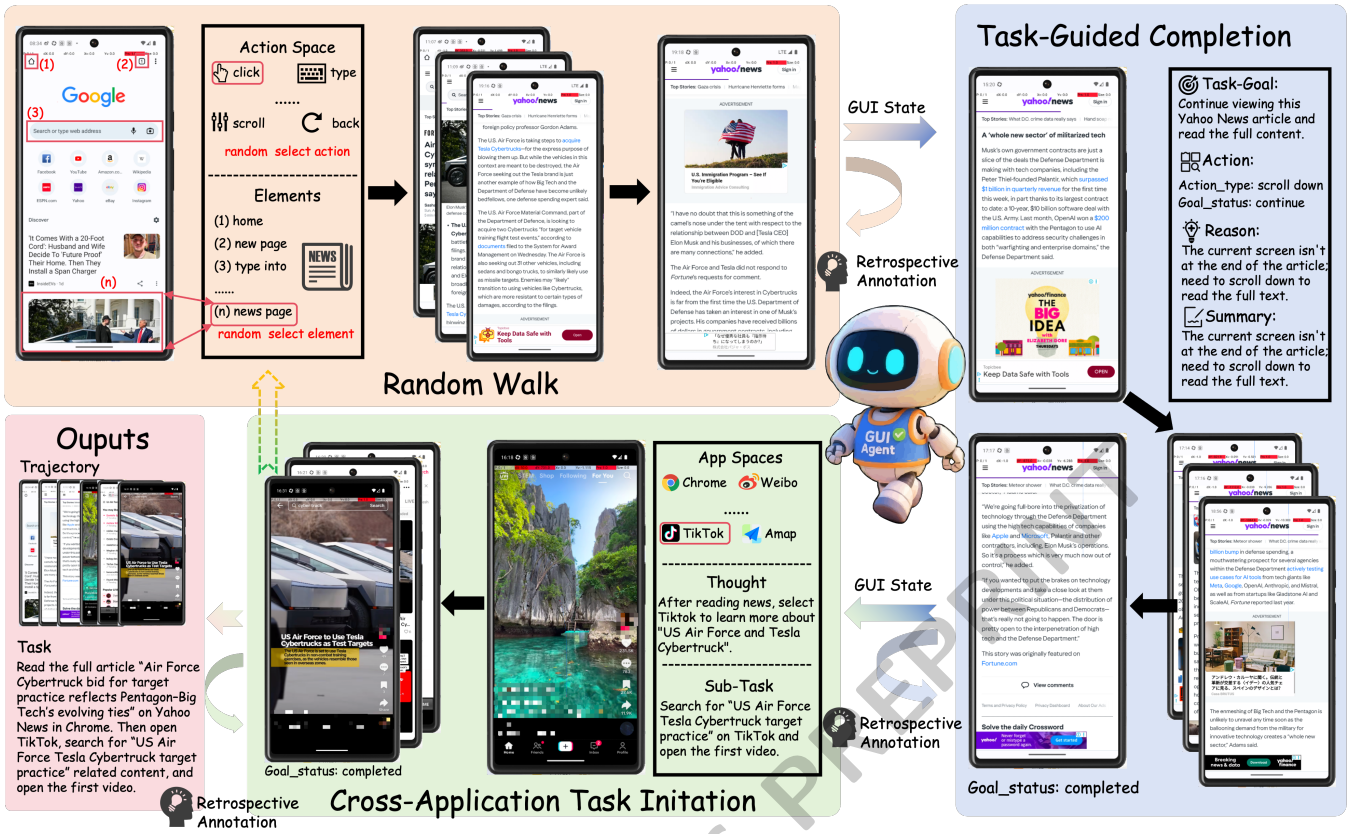

Figure 2: Overview of GUI-ReWalk Framework. Starting from a random app, GUI-ReWalk performs **Random Walk** by selecting actions and interacting with elements step by step; it then transitions to **Task-Guided Completion** to complete minimal-step tasks forming a stride, followed by **Cross-Application Task Initiation** to propose and execute new tasks in related apps. After each sub-stage, **Retrospective Annotation** records executed actions and GUI states. This cycle repeats across multiple strides to generate complete trajectories and overall task objectives.

while fluidly transitioning between exploratory and goal-directed behaviors. When the framework encounters a dead end that prevents task completion, it invokes a reflective reasoning process to revise the original objectives, leveraging both the initial goals and the interaction history, to formulate new, relevant, and executable targets. This capability enables the system to recover progress, thereby ensuring the continuity and robustness of task execution.

3.2 Random Walk

The random walk phase reflects human exploration and boundary probing when interacting with unfamiliar interfaces. Inspired by open-world agents that acquire general skills through extensive exploration [Liu *et al.*, 2025], we simulate this stage as trial-and-error interaction without a fixed objective. GUI-ReWalk models this behavior as a Markov chain:

$$\mathcal{M}_r = (\mathcal{S}, \mathcal{A}^r, P^r),$$

where $\mathcal{S}$ denotes the state of the GUI environment, and $\mathcal{A}^r$ denotes the primitive GUI actions available in this state. The state transition probability of the random walk is defined as

$$P^r(s_{t+1} \mid s_t) = \sum_{a_t^r} P^r(s_{t+1} \mid s_t, a_t^r)\, P^r(a_t^r \mid s_t),$$

where $s_t \in \mathcal{S}$ and $a_t^r \in \mathcal{A}^r$ denote the state and accessible actions at time step t. During the exploration phase, the action policy is set to a uniform distribution $P^r(a_t^r \mid s_t) = \frac{1}{|\mathcal{A}^r|}$ to maximize state-space coverage and emulate the chaotic probing behavior commonly observed in human users. After this, GUI-ReWalk randomly chooses an executable UI element for the selected action. For input-type actions, such as typing, GUI-ReWalk leverages the LLM to generate context-appropriate text. As the trajectory stride extends over multiple iterations, the length of the random walk is gradually reduced to better reflect the natural shift from broad exploration to focused interaction observed in real human behavior.

3.3 Task-guided Completion

After exploring the environment, human users typically form explicit goals and act purposefully. GUI-ReWalk models the task-guided completion phase as a goal-constrained Markov decision process [Kaelbling *et al.*, 1998]. In this process, we

first use the LLM to infer a high-level task goal from the terminal exploration state.

$$g = \Phi_{\text{LLM}}(s_{T_r}), \quad s_{T_r} \in \mathcal{S},$$

where Φ_{LLM} is the LLM goal inference function, T_r is the terminal time step of the random walk. Then the task-guided completion phase is formalized as:

$$\mathcal{M}_g = (\mathcal{S}, \mathcal{A}^g, P^g, \mathcal{R}^g, \pi),$$

The state transition probability of task-guided completion captures how humans act purposefully once they have a clear goal in mind. It can be formulated as:

$$P^t(s_{t+1} \mid s_t) = \sum_{a_t^g} P^t(s_{t+1} \mid s_t, a_t^g)\, \pi(a_t^g \mid s_t),$$

where $a_t^g \in \mathcal{A}^g$ and $\pi(a_t^g \mid s_t)$ is the action policy based on the LLM. $\pi(a_t^g \mid s_t)$ selects the next action based on the current state and the intended goal. When an action cannot be executed within the current environment, $\pi(a_t^g \mid s_t)$ engages a reflective reasoning process to revise the goal g, ensuring that the updated objective remains both relevant and feasible for continued task execution. To reflect the sparsity of meaningful task completion signals, we define a sparse reward [Andrychowicz *et al.*, 2017; Schaul *et al.*, 2015] function as:

$$\mathcal{R}^g = \begin{cases} r_{\text{succ}} = 1, & \text{if } s \in S^g, \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{S}^g = \mathcal{S} \times \mathcal{G}$ is the task-conditioned state space and $\mathcal{G}$ is the goal space. The prompts for Φ_{LLM} is shown in the **Appendix**.

3.4 Cross-application Task Initiation

Similar to the goal inference in the task-guided completion phase, GUI-ReWalk leverages the LLM to analyze the trajectory and annotations of the current stride E_i. Based on this analysis, it generates a semantically related cross-application goal to initiate the next stride:

$$G_{i+1} = \Pi_{\text{LLM}}(s_i),$$

where s_i denotes the final state of the i-th stride, and Π_{LLM} is the goal-generation policy implemented by the LLM. The inferred goal G_{i+1} is then used to initialize the next stride $\mathcal{L}_{i+1}$. Upon switching to a new application, GUI-ReWalk re-enters the process, performing a random walk, task-guided completion, and retrospective annotation, thereby constructing a new stride that continues the multi-application trajectory.

This hierarchical orchestration mirrors human multi-application workflows, where users frequently transition from completing one task to initiating another related task across different applications. It maintains the same alternation between chaotic exploration and goal-directed execution, while ensuring semantic continuity across strides to form coherent, multi-application trajectories.

3.5 Task Recovery

Task recovery addresses scenarios where users deviate from their intended trajectory due to errors, ambiguous goals, or unforeseen interface dynamics. Inspired by MIRROR's intra- and inter-reflection mechanisms for optimized reasoning in tool learning [Guo *et al.*, 2025], GUI-ReWalk models this recovery process as an adaptive replanning mechanism built on the interplay between state monitoring and LLM-driven reasoning.

Formally, when the agent detects that the current trajectory $\tau = \{s_t, a_t\}_{t=1}^{T}$ fails to progress toward the inferred goal g, a recovery trigger is activated:

$$\Omega(s_t, g) = \begin{cases} 1, & \text{if progress towards } g \text{ stalls or repeat,} \\ 0, & \text{otherwise.} \end{cases}$$

Once activated, we use the LLM to re-analyze the current environment to update or refine the task objective:

$$g' = \Psi_{\text{LLM}}(s_{t'}, g),$$

with Ψ_{LLM} denoting the goal-revision function. This allows the agent to dynamically adapt its task representation when the original goal g becomes infeasible or underspecified.

The subsequent execution continues under the revised policy $\pi'(a \mid s, g')$, ensuring that the trajectory realigns with a coherent objective. This recovery loop effectively captures human-like resilience in digital environments, enabling GUI-ReWalk to handle interruptions, erroneous actions, and semantic drift robustly. By incorporating task recovery, the framework closes the loop between exploration, goal-directed execution, and error correction, thereby achieving more reliable and human-aligned multi-application task automation.

4 Experiments and Results

We train GUI-ReWalk on trajectory data generated within our GUI environment, using Qwen-2.5VL-7B as the base model. The evaluation is conducted from two perspectives: grounding and navigation. For grounding, the full controllability of the GUI environment enables the construction of a large-scale dataset for model training. For navigation, we apply LLM-based automated filtering and trajectory scoring to select high-quality samples for supervised fine-tuning (SFT).

4.1 Grounding

We evaluate the grounding capability of GUI-ReWalk on several publicly available benchmarks, including Screenspot-Pro [Li *et al.*, 2025], OSWorld-G [Xie *et al.*, 2025b], and UI-Vision [Nayak *et al.*, 2025]. These benchmarks share a common focus on fine-grained GUI grounding, requiring models to accurately localize target interface elements in complex, real-world desktop or application screenshots based on natural language instructions. The overall results are reported in Table 1, with a detailed breakdown provided in the **Appendix**.

ScreenSpot-Pro targets professional software with high-resolution interfaces, spanning domains such as CAD, programming, creative design, scientific computing, office applications, and operating systems. It emphasizes grounding

Model	Screenspot-Pro	OSWorld-G	UI-Vision
GPT-4o [OpenAI, 2024]	0.8	–	1.4
SeeClick-9.6B [Cheng *et al.*, 2024]	1.1	–	5.4
OS-Atlas-7B [Wu *et al.*, 2024]	18.9	22.7	9.0
UGround-7B [Gou *et al.*, 2025]	16.5	36.4	12.9
UI-TARS-7B [Qin *et al.*, 2025]	**35.7**	**47.5**	**17.6**
Qwen2.5-VL-7B [Bai *et al.*, 2025]	20.8	16.8	3.7
GUI-ReWalk-7B (ours)	35.1	27.5	5.9

Table 1: Performance comparison on grounding datasets. The reported scores represent the average performance across all sub-tasks within each benchmark.

Model	AndroidControl-Low		AndroidControl-High		GUI-Odyssey	
	Type Acc.	SR	Type Acc.	SR	Type Acc.	SR
GPT-4o [OpenAI, 2024]	74.3	19.4	66.3	20.8	34.3	3.3
SeeClick-9.6B [Cheng *et al.*, 2024]	**93.0**	75.0	82.9	59.1	71.0	53.9
OS-Atlas-7B [Wu *et al.*, 2024]	93.6	85.2	**85.2**	**71.2**	**84.5**	62.0
OS-Genesis-7B [Sun *et al.*, 2025]	90.7	74.2	66.2	44.5	–	–
Qwen2.5-VL-7B [Bai *et al.*, 2025]	91.8	85.0	70.9	69.8	59.5	46.3
GUI-ReWalk-7B (ours)	91.7	**96.3**	73.1	66.2	69.6	**64.2**

Table 2: Comparison of models on navigation benchmarks. "Type Acc." denotes type accuracy, and "SR" denotes success rate.

in visually complex environments, where dense and heterogeneous iconography poses substantial challenges. As shown in Table 1, with 100k generated grounding data, GUI-ReWalk improves upon Qwen2.5-VL-7B by 14.3.

OSWorld-G consists of fine-grained tasks that closely simulate authentic computer usage, requiring text matching, element recognition, layout understanding, precise manipulation, and refusal handling. It provides a holistic evaluation of grounding in real-world digital environments. GUI-ReWalk yields an improvement of 10.7 over Qwen2.5-VL-7B.

UI-Vision is a license-permissive benchmark designed for desktop agents. It evaluates grounding performance across diverse and fine-grained tasks in realistic desktop environments, offering a comprehensive assessment of practical grounding capabilities. GUI-ReWalk achieves an improvement of 2.2 compared with Qwen2.5-VL-7B.

GUI-ReWalk delivers consistent improvements in grounding performance across professional software, realistic desktop tasks, and fine-grained computer-use scenarios. Compared with general-purpose vision–language models of the same scale (e.g., Qwen2.5-VL-7B), GUI-ReWalk demonstrates significantly stronger capabilities in recognizing dense iconography, understanding complex layouts, and grounding actions within diverse GUI contexts. Relative to other GUI-specialized models at the 7B scale, GUI-ReWalk shows greater robustness and adaptability, benefiting from its systematic trajectory generation pipeline. Moreover, the framework proves to be highly data-efficient, achieving substantial gains with moderate training data while retaining scalability for larger settings. These results establish GUI-ReWalk as a more reliable and generalizable solution among models of comparable size, highlighting its potential to advance grounding in practical human–computer interaction tasks.

4.2 Navigation

To evaluate the multi-step decision-making capability of our proposed method, GUI-ReWalk, we conduct experiments on several publicly available benchmarks, including AndroidControl [Li *et al.*, 2024] and GUI-Odyssey [Lu *et al.*, 2024]. The overall results are summarized in Table 2, with detailed analyses provided in the Appendix.

AndroidControl is a static offline benchmark designed to evaluate UI comprehension, task decomposition, and action planning under both low-level and high-level task instructions. Compared with Qwen2.5-VL-7B, GUI-ReWalk achieves consistent improvements across both evaluation metrics. Specifically, on low-level tasks, GUI-ReWalk maintains roughly flat type accuracy and boosts step success rate by 11.3. On high-level tasks, it further achieves gains of 2.2 in type accuracy. These results highlight the model's superior ability in hierarchical planning and abstraction.

GUI-Odyssey provides complementary offline evaluation tasks that emphasize structured reasoning and robust action planning in controlled environments. On this benchmark, GUI-ReWalk outperforms Qwen2.5-VL-7B by 10.1 in type accuracy and 17.9 in step success rate, further validating the model's effectiveness in offline multi-step decision-making and complex task decomposition.

To further assess the agent's capability in handling open-ended and complex real-world tasks, we conducted a comprehensive evaluation on the challenging OSWorld benchmark. As presented in Table 5, GUI-ReWalk-7B achieves a success rate of 4.7%, effectively matching the performance of the significantly larger Qwen2.5-VL-32B model. This represents a

Stride Number	AndroidControl-Low		AndroidControl-High		GUI-Odyssey	
	Type Acc.	Value Acc.	Type Acc.	Value Acc.	Type Acc.	Value Acc.
Stride = 1	91.7	31.5	73.1	30.6	69.6	33.0
Stride = 2	91.9	32.1	72.7	38.6	70.8	39.4
Stride = 3	92.0	32.3	72.5	37.9	72.2	39.7

Table 3: Ablation study of trajectory stride number on GUI-ReWalk. "Type Acc." denotes type accuracy, and "Value Acc." denotes value accuracy.

Environments	**Action**	**Definition**	**Mobile Rate**	**Desktop Rate**
Shared	Click(x, y)	Clicks at coordinates (x, y).	61.67%	78.49%
	Scroll(direction)	Scrolls the screen with specified direction.	9.31%	1.03%
	Drag(x1,y1, x2,y2)	Drags from (x1, y1) to (x2, y2).	0.05%	1.22%
	Type(content)	Types the specified content.	9.19%	4.00%
	Wait()	Waits for screen update.	3.14%	0.91%
	Completed()	Marks the task as finished.	7.79%	5.62%
	Infeasible()	Marks the task as cannot be done.	0.56%	1.68%
Mobile	Launch(app)	Opens the specified app.	7.53%	-
	LongPress(x, y)	Long presses at (x, y).	0.21%	-
	PressBack()	Presses the "back" button.	0.32%	-
	PressHome()	Presses the "home" button.	0.16%	-
	PressEnter()	Presses the "enter" key.	0.07%	-
Desktop	HotKey(key)	Performs the specified hotkey.	-	0.59%
	LeftDouble(x, y)	Double-clicks at (x, y).	-	4.33%
	RightSingle(x, y)	Right-clicks at (x, y).	-	2.13%

Table 4: Unified action space for different environments.

Model	Parameters	Success Rate
Qwen2.5-VL	72B	5.0%
Qwen2.5-VL	32B	3.9%
Qwen2.5-VL (Base)	7B	1.4%
GUI-ReWalk (Ours)	**7B**	**4.7%**

Table 5: Performance comparison on the **OSWorld benchmark**. GUI-ReWalk-7B matches the 72B model and significantly outperforms its 7B base.

remarkable leap over the base Qwen2.5-VL-7B model, which scores only 1.4%—translating to a nearly 3× improvement in task completion. Such results compellingly demonstrate that our high-quality, reasoning-dense synthetic data can effectively substitute for sheer parameter scale, enabling efficient 7B-scale models to rival the capabilities of 32B models and remain competitive with 72B counterparts in complex, open-ended environments.

To investigate the impact of trajectory stride on model performance, we conduct an ablation study by varying the stride number during training. As summarized in Table 3, a single stride yields suboptimal grounding and navigation due to limited diversity. Increasing to two strides substantially improves performance by capturing multi-step dependencies. However, three strides offer marginal gains and may introduce noise from redundant sub-trajectories.

Overall, GUI-ReWalk significantly improves multi-step decision-making by balancing low-level operations (AndroidControl) and high-level planning (GUI-Odyssey). By capturing the hierarchical nature of human-computer interaction, our synthetic trajectories enable the 7B model to clearly outperform its baseline. This demonstrates that well-structured data can effectively offset model scale limitations, offering a scalable path toward generalizable GUI agents.

5 Conclusion

In this work, we introduced **GUI-ReWalk**, a reasoning-enhanced framework for synthesizing realistic and diverse GUI interaction trajectories. By unifying stochastic exploration with goal-directed reasoning, GUI-ReWalk captures both the long-tail variability and the structured intent of human-computer interactions. Its multi-stride design enables the construction of long-horizon workflows spanning multiple applications, offering a closer reflection of real-world usage patterns than prior datasets. Extensive evaluations show that training models on GUI-ReWalk yields broader interaction coverage, higher trajectory entropy, and more faithful representations of user intent across diverse benchmarks. Beyond providing a scalable data generation pipeline, GUI-ReWalk underscores the importance of reflective reasoning, error recovery, and platform diversity in advancing GUI agent research, paving the way toward next-generation agents that are both resilient and capable of real-world automation at scale.

Contribution Statement

The authors marked with an asterisk (*) contributed equally to this work.

References

[Andrychowicz *et al.*, 2017] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *NeurIPS 2017*, 30, 2017.

[Anthropic, 2024] Anthropic. Developing a computer use model, 2024.

[Bai *et al.*, 2025] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[Bonatti *et al.*, 2024] Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Jang, and Zack Hui. Windows agent arena: Evaluating multi-modal os agents at scale, 2024.

[Chen *et al.*, 2025] Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, Tianshuo Zhou, Yue Yu, Chujie Gao, Qihui Zhang, Yi Gui, Zhen Li, Yao Wan, Pan Zhou, Jianfeng Gao, and Lichao Sun. Gui-world: A video benchmark and dataset for multimodal gui-oriented understanding, 2025.

[Cheng *et al.*, 2024] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents, 2024.

[Deng *et al.*, 2023] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *NeurIPS 2023*, volume 36, pages 28091–28114. Curran Associates, Inc., 2023.

[Dobrica, 2022] Liliana Dobrica. Robotic process automation platform uipath. *Commun. ACM*, 65(4):42–43, March 2022.

[Gao *et al.*, 2025] Yifei Gao, Junhong Ye, Jiaqi Wang, and Jitao Sang. Websynthesis: World-model-guided mcts for efficient webui-trajectory synthesis, 2025.

[Gou *et al.*, 2025] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *ICLR 2025*, 2025.

[Guo *et al.*, 2025] Zikang Guo, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. Mirror: Multi-agent intra- and inter-reflection for optimized reasoning in tool learning. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 117–125. International Joint Conferences on Artificial Intelligence Organization, 8 2025. Main Track.

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

[Li *et al.*, 2024] Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *NeurIPS 2024*, volume 37, pages 92130–92154. Curran Associates, Inc., 2024.

[Li *et al.*, 2025] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use, 2025.

[Liu *et al.*, 2018] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration, 2018.

[Liu *et al.*, 2025] Shunyu Liu, Yaoru Li, Kongcheng Zhang, Zhenyu Cui, Wenkai Fang, Yuxuan Zheng, Tongya Zheng, and Mingli Song. Odyssey : Empowering minecraft agents with open-world skills. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 187–195. International Joint Conferences on Artificial Intelligence Organization, 8 2025. Main Track.

[Lu *et al.*, 2024] Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices, 2024.

[Lù *et al.*, 2024] Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue, 2024.

[Memon *et al.*, 2003] A. Memon, I. Banerjee, N. Hashmi, and A. Nagarajan. Dart: a framework for regression testing "nightly/daily builds" of gui applications. In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.*, pages 410–419, 2003.

[Nayak *et al.*, 2025] Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A. Rodriguez, Montek Kalsi, Rabiul Awal, Nicolas Chapados, M. Tamer Özsu, Aishwarya Agrawal, David Vazquez, Christopher Pal, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction, 2025.

[OpenAI, 2024] OpenAI. Gpt-4 technical report, 2024.

[Qin *et al.*, 2025] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai,

Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. Ui-tars: Pioneering automated gui interaction with native agents, 2025.

[Rawles *et al.*, 2025] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents, 2025.

[Schaul *et al.*, 2015] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *ICML 2015*, pages 1312–1320. PMLR, 2015.

[Shi *et al.*, 2017] Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In Doina Precup and Yee Whye Teh, editors, *ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR, 06–11 Aug 2017.

[Sun *et al.*, 2025] Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. Os-genesis: Automating gui agent trajectory construction via reverse task synthesis, 2025.

[Tan *et al.*, 2024] Weihao Tan, Ziluo Ding, Wentao Zhang, Boyu Li, Bohan Zhou, Junpeng Yue, Haochong Xia, Jiechuan Jiang, Longtao Zheng, Xinrun Xu, Yifei Bi, Pengjie Gu, Xinrun Wang, Börje F. Karlsson, Bo An, and Zongqing Lu. Towards general computer control: A multimodal agent for red dead redemption II as a case study. In *ICLR 2024 Workshop on LLM Agents*, 2024.

[Tang *et al.*, 2025] Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. Agent kb: Leveraging cross-domain experience for agentic problem solving, 2025.

[Wu *et al.*, 2024] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: A foundation action model for generalist gui agents, 2024.

[Wu *et al.*, 2025] Penghao Wu, Shengnan Ma, Bo Wang, Jiaheng Yu, Lewei Lu, and Ziwei Liu. Gui-reflection: Empowering multimodal gui models with self-reflection behavior, 2025.

[Xia *et al.*, 2024] Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Agentless: Demystifying llm-based software engineering agents, 2024.

[Xie *et al.*, 2024] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *NeurIPS 2024*, volume 37, pages 52040–52094. Curran Associates, Inc., 2024.

[Xie *et al.*, 2025a] Jingxu Xie, Dylan Xu, Xuandong Zhao, and Dawn Song. Agentsynth: Scalable task generation for generalist computer-use agents, 2025.

[Xie *et al.*, 2025b] Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis, 2025.

[Xu *et al.*, 2025] Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction, 2025.

[Yan *et al.*, 2023] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, Zicheng Liu, and Lijuan Wang. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation, 2023.

[Yang *et al.*, 2023] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions, 2023.

[Yang *et al.*, 2025] Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Caiming Xiong, and Junnan Li. Gta1: Gui test-time scaling agent, 2025.

[Yao *et al.*, 2022] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *NeurIPS 2022*, volume 35, pages 20744–20757. Curran Associates, Inc., 2022.

[Zhang *et al.*, 2025a] Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Building generalized gui agents by learning from multimodal web tutorials, 2025.

[Zhang *et al.*, 2025b] Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large language model-brained gui agents: A survey, 2025.

[Zhou *et al.*, 2024] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024.