OpAgent: Operator Agent for Web Navigation

Yuyu Guo* Wenjie Yang* Siyuan Yang* Ziyang Liu* Cheng Chen Yuan wei
Yun Hu Yang Huang Guoliang Hao Dongsheng Yuan Jianming Wang Xin Chen
Hang Yu† Lei Lei Peng Di

Ant Group

https://github.com/codefuse-ai/OpAgent
https://huggingface.co/codefuse-ai/OpAgent

Abstract

To fulfill user instructions, autonomous web agents must contend with the inherent complexity and volatile nature of real-world websites. Conventional paradigms predominantly rely on Supervised Fine-Tuning (SFT) or Offline Reinforcement Learning (RL) using static datasets. However, these methods suffer from severe distributional shifts, as offline trajectories fail to capture the stochastic state transitions and real-time feedback of unconstrained wide web environments. In this paper, we propose a robust Online Reinforcement Learning WebAgent, designed to optimize its policy through direct, iterative interactions with unconstrained wide websites. Our approach comprises three core innovations: 1) Hierarchical Multi-Task Fine-tuning: We curate a comprehensive mixture of datasets categorized by functional primitives—Planning, Acting, and Grounding—establishing a Vision-Language Model (VLM) with strong instruction-following capabilities for Web GUI tasks. 2) Online Agentic RL in the Wild: We develop an online interaction environment and fine-tune the VLM using a specialized RL pipeline. We introduce a Hybrid Reward Mechanism that combines a ground-truth-agnostic WebJudge for holistic outcome assessment with a Rule-based Decision Tree (RDT) for progress reward. This system effectively mitigates the credit assignment challenge in long-horizon navigation. Notably, our RL-enhanced model achieves a 38.1% success rate (pass@5) on WebArena, outperforming all existing monolithic baselines. 3) Operator Agent: We introduce a modular agentic framework, namely **OpAgent**, orchestrating a Planner, Grounder, Reflector, and Summarizer. This synergy enables robust error recovery and self-correction, elevating the agent's performance to a new State-of-the-Art (SOTA) success rate of **71.6%**.

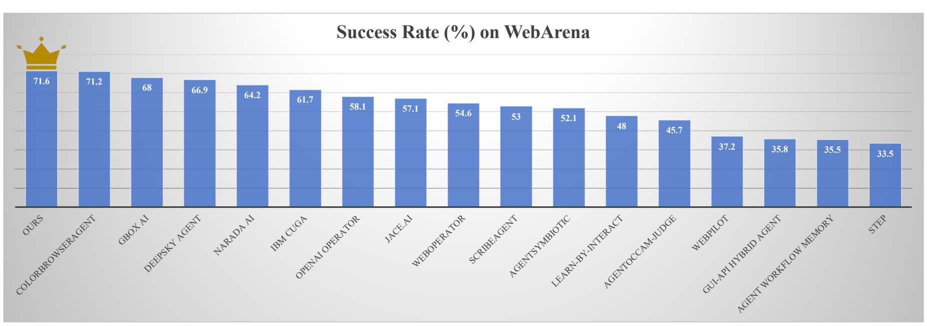

Figure 1: Our proposed OpAgent achieves a new state-of-the-art (SOTA) success rate of 71.6% on the WebArena benchmark.

*Equal Contribution.
†Correspondence to: Hang Yu <hyu.hugo@antgroup.com>.

1 Introduction

The rapid advancement of Large Language Models (LLMs) Yang et al. (2025a); OpenAI (2023); Team (2025) and Vision-Language Models (VLMs) Bai et al. (2025a); Wang et al. (2024; 2025b) has brought the development of autonomous agents—capable of navigating complex environments to fulfill user instructions—within practical reach. Previous research has extensively explored autonomous agents Wang et al. (2025a); Hong et al. (2024) across various domains, spanning PC Gonzalez-Pumariega et al. (2025); Liu et al. (2025), mobile Xu et al. (2025a); Ye et al. (2025), and web Zhou et al. (2026a); Shen et al. (2025b) environments.

Compared to desktop and mobile interfaces, the web presents three unique hurdles:

- **Unstructured Complexity:** The underlying DOM tree is often orders of magnitude larger and more cluttered than desktop accessibility trees, laden with redundant or hidden metadata.
- **Temporal Volatility:** Websites exhibit extreme dynamicity, where asynchronous updates and ephemeral content render static, offline datasets rapidly obsolete.
- **Interaction Latency and Implicit Logic:** Many web operations require real-time feedback to verify success (*e.g.*, hover-triggered menus), creating an interactive closed-loop that offline learning is inherently unable to simulate.

Despite these challenges, conventional paradigms predominantly rely on offline learning, which suffers from two fundamental limitations. Regarding *unstructured complexity*, existing methods Deng et al. (2023) heavily depend on text-based HTML or DOM tree parsing. Such representations are inherently fragile and sensitive to the immense noise found in modern websites, often overwhelming the agent's reasoning capacity. Regarding *volatility and implicit logic*, previous "static" approaches Gou et al. (2025); Gu et al. (2025)—whether based on SFT or Offline RL—inevitably encounter severe **distributional shifts**. Since offline trajectories are pre-recorded, they fail to capture stochastic state transitions or provide the exploration-driven feedback loop necessary for mastering implicit interaction logic. Consequently, without the benefit of real-time trial-and-error, agents trained offline struggle to recover from execution failures or adapt to the fluid nature of live web environments.

To bridge the gap between static training and dynamic execution, we propose **OpAgent**, a robust framework designed for autonomous web navigation. Our approach systematically tackles the aforementioned hurdles through three core strategies:

First, to navigate the **unstructured complexity** of the web, we transcend traditional text-centric paradigms. Instead, we leverage a Vision-Language Model (VLM) Bai et al. (2025b;a) to directly perceive visual signals, mimicking human-centric web interaction. This allows the agent to capitalize on spatial semantics and visual hierarchies that are more stable and informative than volatile HTML code. To forge a robust foundational policy, we curate a multi-task dataset categorized into three functional primitives: **Grounding, Planning, and Acting**, followed by a weighted multi-task Supervised Fine-Tuning (SFT) Guo et al. (2018); Liu et al. (2024) process.

Second, to overcome **temporal instability** and **implicit logic**, we introduce an Online Agentic Reinforcement Learning framework. Unlike static training, our framework enables the agent to interact with live websites in real-time, effectively mitigating distributional shifts through autonomous exploration. We optimize the policy using a **Hybrid Reward Mechanism**, which synergizes a high-level VLM-based **WebJudge** Xue et al. (2025) for holistic outcome verification with a **Rule-based Decision Tree (RDT)** for granular progress rewards.

Finally, we orchestrate these optimized capabilities within a modular architecture. By coordinating a specialized **Planner, Grounder, Reflector, and Summarizer**, **OpAgent** achieves sophisticated reasoning and robust self-correction, ensuring high success rates in long-horizon tasks.

Our main contributions are summarized as follows:

- **Hierarchical Multi-Task Skill Acquisition:** We establish a foundational policy by curating a comprehensive dataset focused on *Planning, Grounding, and Acting*. By prioritizing visual perception over brittle HTML parsing, our model effectively tames the unstructured complexity of modern web environments.
- **Agentic Online RL with Hybrid Rewards:** We develop an online RL pipeline that optimizes policies through live web interactions. By synergizing the **WebJudge** evaluator with a **Rule-based Decision Tree (RDT)**, we provide dense, ground-truth-agnostic supervision that effectively alleviates credit assignment challenges in the wild. Our RL-enhanced monolithic model achieves **38.1% (pass@5)** on WebArena, outperforming existing monolithic baselines.
- **Collaborative Architecture and SOTA Performance:** We introduce a modular agentic framework that orchestrates specialized roles for error recovery. **OpAgent** achieves a new State-of-the-Art (SOTA) success rate of **71.6%** on the WebArena benchmark, securing the top position on the leaderboard (January 2026).

2 Related Work

2.1 Autonomous Agents with LLMs and VLMs

The emergence of Large Language Models (LLMs) OpenAI (2023); Yang et al. (2025a) and their vision-language counterparts (VLMs) Bai et al. (2025a); Wang et al. (2024) has revolutionized the development of autonomous agents. Early works focused on text-based reasoning and tool use in sandboxed environments Yao et al. (2023); Schick et al. (2023). Recently, VLM-based agents have demonstrated superior performance in GUI-based tasks by directly perceiving visual signals Wang et al. (2025a); Hong et al. (2024). However, most of these agents are designed for general-purpose interaction and often lack the specialized policy optimization required for the highly dynamic and unstructured web environment.

2.2 Web Navigation Agents

Web navigation is a long-standing challenge in the AI community. Traditional methods predominantly relied on DOM tree parsing and text-centric representations Deng et al. (2023). While effective for simple layouts, they struggle with the noise and complexity of modern websites. To address this, recent works such as WebVoyager He et al. (2024) and SeeAct Zheng et al. (2024) have transitioned to vision-based interaction. Despite their progress, these agents primarily rely on prompt engineering or Supervised Fine-Tuning (SFT) on static datasets. As highlighted in our study, these "static" paradigms suffer from significant distributional shifts when deployed in real-world, wild web environments.

2.3 Reinforcement Learning for Agents

Reinforcement Learning (RL) has been widely explored to enhance agent decision-making. Initial attempts focused on Offline RL, where policies are optimized using pre-recorded expert trajectories Luo et al. (2025); Zhou et al. (2025). However, offline methods are constrained by the quality and coverage of the static datasets. To bridge the gap between training and execution, recent research has begun exploring Online RL in simulated environments with text observation Wei et al. (2025). Our work advances this frontier by implementing an Online Agentic RL framework in the wild with a vision-language model, utilizing a ground-truth-agnostic Hybrid Reward Mechanism to provide dense supervision without the need for oracle trajectories.

2.4 Agentic Architectures and Self-Correction

Complex, long-horizon tasks often require more than a single-step policy. Architectures such as Reflexion Shinn et al. (2023) have introduced self-correction mechanisms where agents reflect on their past errors. In the web domain, multi-agent orchestration has also shown promise in decomposing complex goals into sub-tasks Zhou et al. (2026b). Our Collaborative Agentic Architecture builds upon these concepts by orchestrating a modular team of Planner, Grounder, Reflector, and Summarizer, specifically optimized for the robust error recovery needed in web navigation.

3 Method

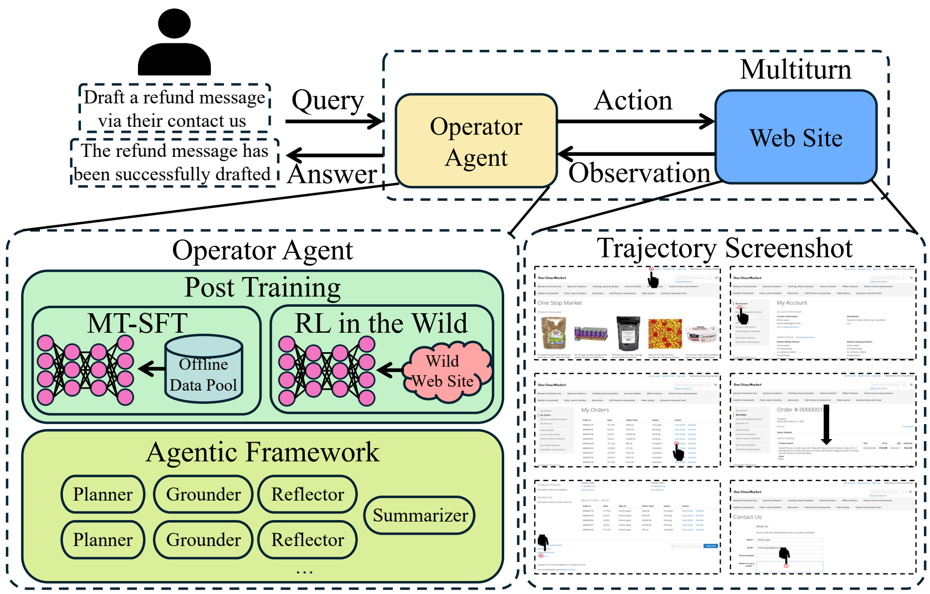

Figure 2: Overall architecture and training pipeline of O p A gen t. (Top) The system facilitates a multi-turn interaction loop where the Operator Agent executes actions and receives observations from live websites to fulfill user queries. (Bottom-Left) The development of the agent follows a hierarchical post-training paradigm: MT-SFT on offline data to establish foundational capabilities, followed by RL in the Wild for adaptive policy optimization in real-world environments. The agentic framework orchestrates modular roles including Planner, Grounder, Reflector and Summarizer. (Bottom-Right) A sample trajectory demonstrates the step-wise execution of a complex refund request task.

As shown in Figure 2, the OpAgent framework operates through a closed-loop interaction between the user, the Operator Agent, and the live web environment. Upon receiving a natural language instruction, the agent iteratively perceives the current visual state of the website, reasons over the multimodal information, and executes a sequence of actions until the task is successfully concluded with an answer.

To empower the agent with robust navigation in unconstrained environments, our training and execution paradigm is structured into three integrated components. First, we establish a cross-modal foundational policy through a **Hierarchical Multi-task Supervised Fine-tuning (MT-SFT)** stage, where the model learns fundamental interaction primitives including planning, grounding, and acting. Second, to bridge the gap between static data and dynamic web environments, we perform **Online Agentic RL in the Wild**, allowing the agent to refine its policy through real-time trial-and-error using a hybrid reward mechanism. Finally, during inference, these capabilities are orchestrated by an OpAgent architecture comprising specialized modules: Planner, Grounder, Reflector, and Summarizer. In the following sections, we elaborate on each component of our framework.

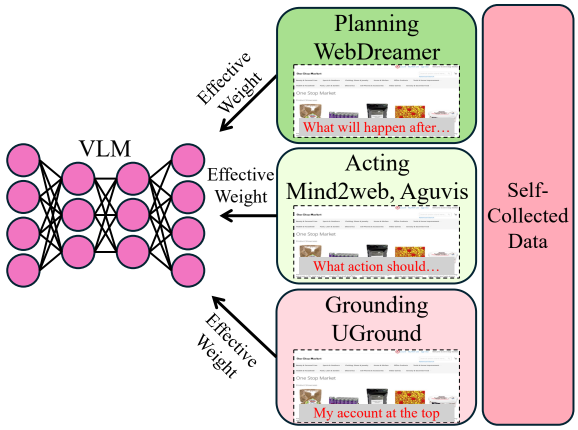

Figure 3: Illustration of the Hierarchical Multi-Task Supervised Fine-tuning (MT-SFT) pipeline. We initialize the VLM by joint training on a diverse mixture of self-collected data, categorized into three functional primitives: (1) **Planning** (via WebDreamer) for high-level goal decomposition and state prediction; (2) **Acting** (via Mind2Web and Aguvis) for low-level action execution; and (3) **Grounding** (via UGround) for spatial element localization. A *Task-specific Effective Weighting* strategy is employed to balance the learning gradients across these heterogeneous tasks, ensuring a robust foundational policy for subsequent RL optimization.

3.1 Multitask Supervised Finetuning with Effective Weight

We categorize the fundamental capabilities of a web agent into three dimensions: **Planning**, **Acting**, and **Grounding**. Specifically, **Planning** involves assessing the functional affordance of UI controls, *i.e.*, anticipating the subsequent page transitions and state changes triggered by a specific interaction. **Acting** entails determining the requisite operation to be performed on the current page to advance the task. Finally, **Grounding** addresses the spatial localization, specifying the precise coordinates or UI elements where the action should be executed. We foster the development of each functional primitive by incorporating targeted open-source datasets. For the *Planning* dimension, we employ WebDreamer Gu et al. (2025) to enhance the model's reasoning about state transitions. For Acting, we rely on the large-scale interaction trajectories from Mind2Web Deng et al. (2023) and Aguvis Xu et al. (2024). Finally, UGround Gou et al. (2025) is utilized to refine the agent's *Grounding* precision through its high-quality spatial annotations. However, a direct concatenation of these heterogeneous datasets poses a significant data imbalance challenge. For instance, while Mind2Web contains only a few thousand steps, UGround reaches the million-scale. Consequently, naive joint training would inevitably cause the model to be dominated by the task with the larger data volume, compromising performance on smaller but crucial tasks. To mitigate this, we introduce a weighting strategy based on the effective number of samples Cui et al. (2019). Let C be the number of different datasets and n_i be the number of samples for dataset $i \in \{1, \ldots, C\}$. We define the hyperparameter β as:

$$\beta = 1 - 10^{-k}, \tag{1}$$

where k is a scaling factor obtained from the experimental configuration. The effective number of samples E_{n_i} for dataset i is calculated as:

$$E_{n_i} = \frac{1 - \beta^{n_i}}{1 - \beta}. \tag{2}$$

To balance the contribution of each task while maintaining gradient stability, we calculate the normalized **effective weight** α_i:

$$\alpha_i = C \cdot \frac{E_{n_i}^{-1}}{\sum_{j=1}^{C} E_{n_j}^{-1}} = C \cdot \frac{(1 - \beta^{n_i})^{-1}}{\sum_{j=1}^{C} (1 - \beta^{n_j})^{-1}}. \tag{3}$$

During the MT-SFT stage, for a batch of size B, the total loss $\mathcal{L}_{SFT}$ is defined as the weighted average of individual sample losses:

$$\mathcal{L}_{SFT} = \frac{1}{B} \sum_{m=1}^{B} \alpha_{dataset(m)} \cdot \left(\frac{1}{T_m} \sum_{t=1}^{T_m} \ell_{m,t} \right), \tag{4}$$

where $\ell_{m,t}$ represents the cross-entropy loss for the t-th token in the m-th sample, T_m is the sequence length, and $\alpha_{dataset(m)}$ is the effective weight corresponding to the dataset category of sample m. This mechanism ensures that the model develops balanced proficiency across Planning, Acting, and Grounding regardless of their original data scales.

3.2 Online Agentic RL in the Wild

Owing to the lack of stable and efficient reinforcement learning (RL) frameworks for web agents, we first developed a specialized browser-based RL infrastructure. We then optimized the agent's policy through a novel **hybrid reward mechanism** that balances outcome-based and process-based supervision.

3.2.1 Hierarchical Infrastructure for Web RL in the Wild

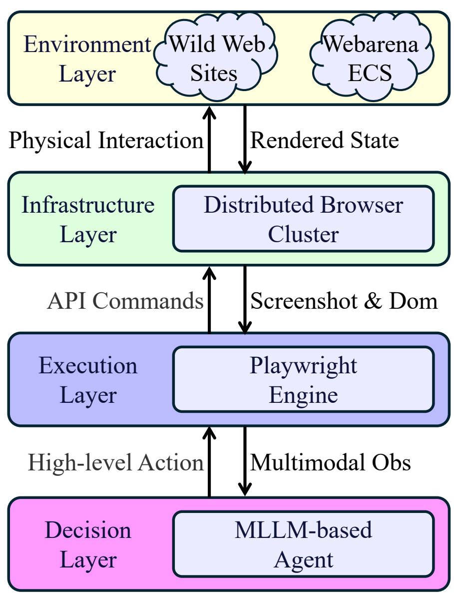

Figure 4: Hierarchical Infrastructure for the Web Agent RL. The system is organized into four functional layers: (1) the **Environment Layer** featuring a hybrid sandbox consisting of self-hosted the open **Wild Web** and **WebArena** on **Alibaba Cloud ECS**; (2) the **Infrastructure Layer** managing a distributed browser cluster for scalable data collection; (3) the **Execution Layer** utilizing a high-concurrency Playwright engine to translate semantic actions into API commands; and (4) the **Decision Layer** where the VLM-based agent performs reasoning and action generation. The solid arrows (left) denote the upward *Action Flow*, while the dashed arrows (right) represent the downward *Observation Flow* of multimodal feedback.

During the Online RL process, the model interacts with the environment throughout the rollout phase, following the dual-stream workflow illustrated in Figure 4. In the action flow, the VLM receives multimodal observations and generates high-level semantic actions in JSON format. Playwright then parses these JSON descriptions into executable API commands and dispatches them to the Distributed Browser Cluster. Within this infrastructure layer, individual browser instances perform the physical interactions on the target websites, including Wild Web sites and WebArena on ECS. Subsequently, in the observation flow, the Browser Cluster renders the updated web states and captures raw data, including screenshots and DOM trees. Playwright then retrieves this raw information, performs data filtering and structuring, and returns a refined multimodal observation to the VLM. This four-tier loop ensures that the agent can collect high-quality trajectories at scale for iterative policy optimization. Furthermore, it is noteworthy that because the Playwright library is not inherently thread-safe, we implemented a dedicated thread pool during the initialization phase to enable asynchronous concurrency. This design effectively avoids the performance overhead of frequent instantiation and destruction of Playwright instances and mitigates potential memory leaks associated with repeated process creation and termination.

3.2.2 Online Agentic RL with Hybrid Reward

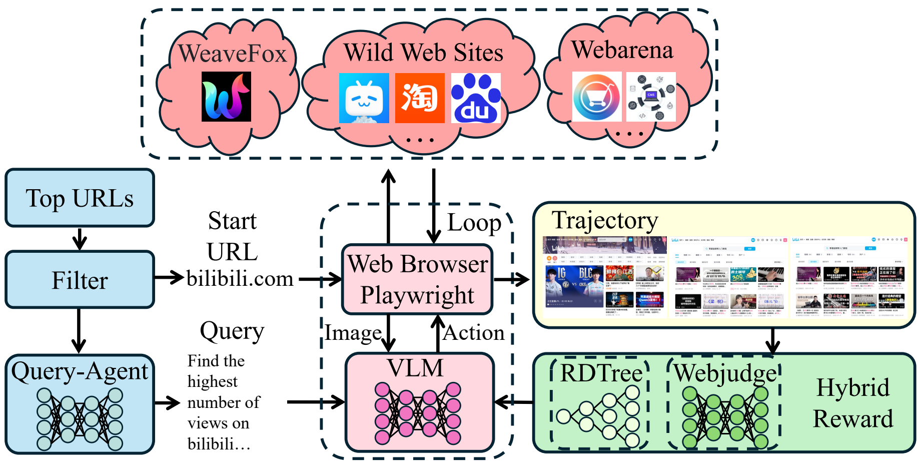

Figure 5: Overview of the OpAgent Training Infrastructure and Reinforcement Learning Loop. The framework consists of three core phases: (1) **Task Generation**, where a Query-Agent synthesizes realistic navigation goals on filtered top URLs; (2) **Interactive Rollout**, where the VLM-based agent interacts with a hybrid environment (including WeaveFox[1], unconstrained Wild Web sites, and the self-hosted WebArena) via a high-concurrency Playwright engine; and (3) **Hybrid Reward Evaluation**. The reward system integrates an **RDTree** (Rule-based Decision Tree) to derive process-based rewards for intermediate steps, and **Webjudge**, which assesses visual trajectory screenshots to provide a holistic success score.

To facilitate reinforcement learning in unconstrained environments, we first curate a collection of top-tier URLs across various domains and leverage a Large Language Model (LLM) to synthesize a diverse set of realistic user queries. During the rollout phase, the process begins with the Playwright engine navigating to the designated starting URL and transmitting the initial visual state to the Vision-Language Model (VLM). The agent then interacts with the website based on the generated query, as shown in Figure 5. Notably, for multi-turn interactions, we adopt a text-centric history strategy: only the current visual observation (screenshot) is provided to the model at each timestep. While the history of previous reasoning and actions is preserved in textual format, all prior images are discarded. This design choice is informed by our empirical findings that current VLMs are prone to visual hallucinations when processing sequences of multiple images, which can significantly degrade the accuracy of decision-making. The rollout process for each trajectory continues until the model signals task completion or reaches a predefined maximum step limit, after which the collected sequence is sent for reward evaluation. We employ Group Relative Policy Optimization (GRPO) DeepSeek-AI (2025); Shao et al. (2024) to optimize the model. To prevent the issue of entropy collapse, we implemented the *KL Cov* strategy Cui et al. (2025). Unlike traditional global KL regularization, this approach selectively applies KL divergence constraints only to tokens that

[1] https://github.com/weavefox

exhibit a high covariance between their advantage values and logits, thereby maintaining policy diversity during training.

To provide comprehensive supervision for policy optimization, we develop a multi-faceted reward function comprising structural and quality-based components. First, a Format Reward is employed to enforce structural compliance, ensuring the presence of a reasoning process (Chain-of-Thought) and the syntactic correctness of the output format. Beyond basic formatting, we evaluate the quality of the navigation trajectory from two distinct perspectives: *i.e.*, Outcome-based Evaluation and Process-based Supervision.

Algorithm 1 Rule-based Decision Tree for Process Reward

Require: Action a_t, Previous State s_{t-1}, Current State s_t, User Query q
Ensure: Step Reward r_t

if Execution Failed (a_t) **then**
    return $r_{penalty}$
end if
if URL Changed (s_{t-1}, s_t) **then** ▷ Significant state transition
    return 0
else if Coordinates on Interactable Element (a_t) **then** ▷ Valid UI affordance
    return 0
else if No Visual Change $(SSIM(s_{t-1}, s_t) == 1.0)$ **then** ▷ Redundant action
    return $r_{penalty}$
else if VLM Judges Progress (s_{t-1}, s_t, q) **then** ▷ Semantic verification
    return 0
else
    return $r_{penalty}$ ▷ Visual noise or irrelevant change
end if

Outcome-based Evaluation (Webjudge): We utilize Webjudge Xue et al. (2025) as a global evaluator to assess task completion. By analyzing the screenshots captured throughout the entire trajectory, Webjudge assigns an outcome reward based on the final visual state and goal fulfillment. Specifically, Webjudge evaluates the quality of each trajectory across three primary dimensions: **Task Completion (Score: -1–5):** This metric assesses the final fulfillment of the user's goal based on a result-oriented rubric. It quantifies whether the agent has successfully resolved the query through its interaction sequence. A score of -1 is specifically reserved for trajectories where the agent encounters insurmountable external obstacles, such as mandatory authentication (login walls) or network connectivity failures. To ensure the purity of the training signal and prevent the model from learning from environmental contingencies, we systematically discard all trajectories with negative scores during the RL training phase. **Action Validity (Score: 1–5):** This score evaluates the precision of element localization. By analyzing the visual variations between consecutive screenshots, Webjudge infers whether the agent's actions correctly targeted the intended UI components. **Trajectory Efficiency (Score: 1–5):** This metric measures the conciseness and logical flow of the interaction path, penalizing redundant steps or circular navigation according to predefined trajectory efficiency standards.

Process-based Supervision (Rule-based Decision Tree): To provide more granular feedback, we implement a process reward that evaluates the functional validity of each step. This mechanism verifies whether the agent's predicted actions (*e.g.*, clicks or types) are executed on effective UI elements with actual affordance, thereby penalizing redundant or invalid interactions. The decision progress is shown in Algorithm 1.

The decision tree mechanism evaluates the validity of an action a_t at state s_t through the following hierarchical checks: Navigation Verification (URL Change): First, the system detects if the action triggered a page transition by comparing the current URL with the previous one. A change in URL implies a successful navigation event, which is considered a valid step (Reward = 0). Affordance Validation (Element Interaction): If the URL remains unchanged, the system verifies if the action (specifically for clicks or hovers) was performed on a valid UI element. It checks whether the action's coordinates (x, y) fall within the bounding box of any interactable element (e.g., buttons, links, inputs) registered in the browser's accessibility tree. Valid interactions are deemed effective (Reward = 0). Redundancy Check (Visual Stagnation): For actions that neither change the URL nor hit a known interactable element, the system checks for visual state changes using the Structural Similarity Index (SSIM). If $SSIM(s_t, s_{t+1}) = 1.0$, indicating the screen is pixel-perfect identical to the previous frame, the action is classified as redundant or invalid (Reward = penalty, e.g., -0.001). Semantic Progress Evaluation (VLM-based Check): If the screen did change visually but failed the previous explicit checks, a generic Visual Language Model (VLM), *e.g.*Qwen-VL-Max, is invoked as a fallback. The VLM compares the pre-action and post-action screenshots given the user's instruction to determine if the visual change represents meaningful progress. If the VLM confirms progress, the step is valid; otherwise, it is penalized.

3.3 Operator Agentic Architecture for Web

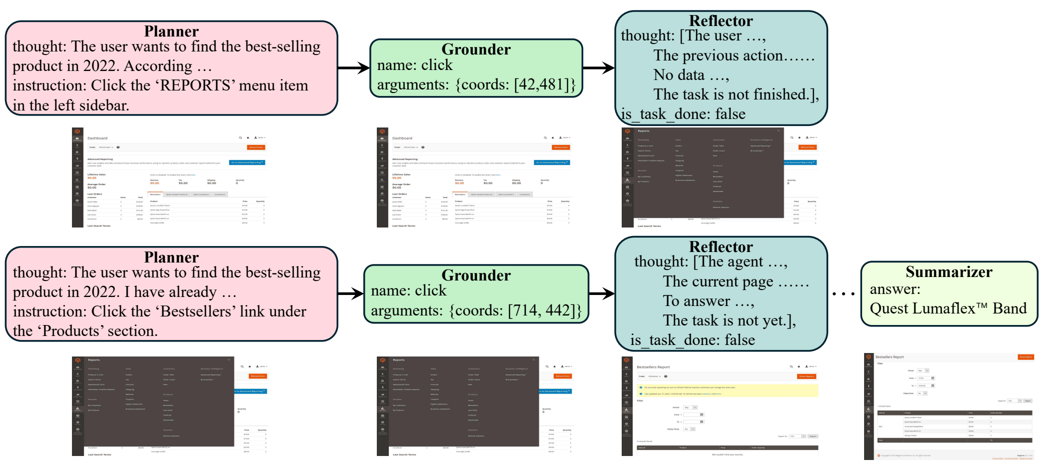

Figure 6: The Iterative Reasoning Pipeline of OpAgent. Our agentic framework decouples complex web navigation into four specialized modules: (1) the **Planner** generates high-level strategic thoughts and instructions; (2) the **Grounder** translates semantic instructions into precise executable actions and coordinates; (3) the **Reflector** evaluates the state transition and monitors task progress; and (4) the **Summarizer** module synthesizes the final answer once the goal is achieved. This loop ensures robust error correction and precise UI interaction.

Due to the inherent complexity of web-based tasks, which often involve dozens of interaction steps, standalone models struggle to achieve satisfactory performance in isolation. At present, an agentic framework equipped with self-reflection and hierarchical task decomposition capabilities remains indispensable for navigating such intricate and long-horizon web environments. Accordingly, our work introduces a sophisticated agentic framework comprising four specialized modules: the Planner, the Grounder, the Reflector, and the Summarizer. The overall pipeline and the specific capabilities of our agentic framework are illustrated in Figure 6 and Table 1, respectively.

Planner: Strategic Decomposition The Planner acts as the strategic core, responsible for decomposing the global *user_query* into atomic, executable steps. It analyzes the current visual state, the pending *todo-list*, and feedback from the Reflector to synthesize a high-level semantic *instruction*. Its key capabilities include:

- **Context-Aware Planning:** Integrating historical trajectories and expert "tips" to dynamically adjust navigation strategies.
- **Adaptive Re-planning:** Utilizing reflection signals to correct erroneous paths or bypass failed interaction attempts.

Grounder: Visual-Action Mapping The Grounder serves as the execution bridge that translates semantic intent into physical interactions. It receives the Planner's instructions and maps them to precise browser coordinates or tool-call parameters. Its key capabilities include:

- **Visual Grounding:** Identifying specific UI elements (*e.g.*, buttons, input fields) from raw screenshots.
- **Precise Actuation:** Generating tool calls (*e.g.*, click, type, scroll) formatted for the underlying browser engine.

Reflector: Introspective Monitoring The Reflector is an introspective module that ensures the reliability of the perception-action loop. It scrutinizes state transitions after each action to verify success and manage information extraction. Its key capabilities include:

- **Factual Verification:** Confirming if an action achieved its intended effect based strictly on visual evidence, thus preventing hallucination-driven progress.
- **Incremental Extraction:** Identifying and recording goal-relevant information into structured *marked notes*.
- **Blocker Detection:** Monitoring for "hard blockers" such as login walls or CAPTCHAs and triggering early termination when necessary.

Summarizer: Evidence Synthesis The Summarizer performs a holistic review of the entire interaction episode. Once the Reflector signals completion or the step limit is reached, it distills the trajectory into a concise final answer. Its key capabilities include:

- **Temporal Fusion:** Synthesizing information across the full sequence of screenshots and collected notes.
- **Goal Assessment:** Providing a final determination of task success and outputting the synthesized result to the user.

Table 1: Roles and Responsibilities within the Modular Agentic Framework.

Module	Core Responsibility	Key Output
Planner	Task decomposition and strategy formulation	Semantic Instruction
Grounder	Mapping semantic intent to UI coordinates	Tool Call
Reflector	Verifying action success and extracting data	Reflection Signal & Notes
Summarizer	Final synthesis of trajectory evidence	Consolidated Answer

4 Experiments

In our experiments, we utilize Qwen2.5-VL-72B-Instruction and Qwen3-VL-32B-Thinking as the primary models for post-training. In OpAgent Architecture, certain sub-agents are further empowered by Gemini-3-Pro to ensure robust performance across diverse web tasks on the WebArena benchmark.

4.1 Multitask Supervised Finetuning with Effective Weight

For the Multi-task Supervised Fine-Tuning (MFT) with Effective Weight, we evaluate our approach on several established offline GUI benchmarks, including operation benchmarks (GUIAct Chen et al. (2025)) and grounding benchmarks (ScreenSpot Li et al. (2025) and ScreenSpot-v2 Wu et al.). The results are shown in Table 2 and Table 3, as seen, our MFT model achieves competitive performance on these benchmark. To be more specific, our model demonstrates superior performance on text-based UI elements, achieving scores of 97.8 on PC text and 94.3 on Web text in the Screenspot V2 benchmark. This stems from our extensive collection of web data, as web environments feature a significantly higher proportion of text-containing controls compared to PC and mobile platforms.

Table 2: Performance comparison on Web-Multi subset of GUIAct benchmark

Method	Backbone	Type EM	Cli.Acc	StepSR
MiniCPM-GUI Chen et al. (2025)	MiniCPM-V	67.0	45.5	47.5
Qwen-GUI Chen et al. (2025)	Qwen-VL	68.9	52.5	46.8
MFT (Ours)	Qwen2.5-VL-7B	83.3	64.4	71.9
MFT (Ours)	Qwen2.5-VL-72B	**84.1**	**67.3**	**73.6**

4.2 Online Agentic RL with Hybrid Reward

To assess the generalization of the online RL-optimized models, we curated 87 Wild Websites and user queries that were not included in the training distribution. We employ Webjudge as an automated evaluator to assign scores based on task completion and trajectory quality. As illustrated in Table 4, the model RL-optimized with Hybrid Reward achieves a substantial improvement of **1.55 points** in the average score compared to the baseline Qwen2.5-VL-72B. Notably, the model is trained through an iterative pipeline. Specifically, we harvest high-quality trajectories (those with high reward scores) generated during the Online RL phase and incorporate them into the Supervised Fine-Tuning (SFT) dataset. This iterative refinement significantly enhances the SFT model's capabilities, ultimately boosting the post-RL performance from 3.09 to 3.56. As illustrated in Figure 7, the optimized model demonstrates a significant performance boost—exceeding **2 points**—across the majority of sub-domains, such as Automotive, News, Education, and Finance.

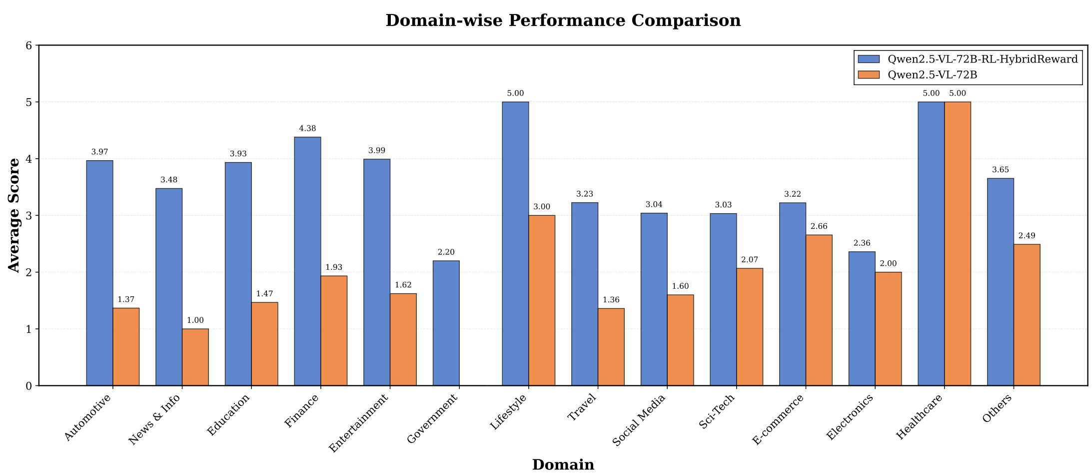

Figure 7: Detailed performance comparison across different sub-domains on Wild Websites. We compare the base Qwen2.5-VL-72B against the RL-tuned Qwen2.5-VL-72B-RL-HybridReward

Beyond the evaluations on "Wild Website" datasets, we further validate the effectiveness of our *HybridReward* method within a controlled and stable virtual environment. Specifically, we employ

Table 3: Performance comparison on Screenspot (v1) and Screenspot-v2 (v2).

	Mobile-Text		Mobile-Icon		Pc-Text		Pc-Icon		Web-Text		Web-Icon		**Avg**	
Method	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2
GUI-Actor-7B Wu et al. (2025)	**94.9**	**96.5**	82.1	84.3	91.8	91.7	80.0	84.1	**91.3**	93.9	85.4	82.3	88.3	89.5
UI-TARS-72B Wang et al. (2025a)	**94.9**	94.8	82.5	**86.3**	89.7	91.2	**88.6**	**87.9**	88.7	91.5	85.0	**87.7**	88.4	90.3
MFT (Ours)	93.4	95.7	**82.8**	85.9	**96.4**	**97.8**	84.3	87.3	90.8	**94.3**	**89.4**	84.4	**89.2**	**91.3**

Table 4: Performance comparison on Wild Websites. Zero means RL without SFT phase. The baseline is Qwen2.5-VL-72B-Instruct.

Model	**Max**	**Min**	**Avg***	**Valid**	**InValid**
Baseline	5.0	1.0	2.01	75	12
RL-HybridReward-Zero	5.0	0.0	3.09	70	17
RL-HybridReward	5.0	1.0	3.56	77	10

* Average score is calculated on a 5-point scale, excluding trajectories with negative scores, *i.e.*, InValid Sample.
Valid: Samples with score ≥ 1; **InValid**: Samples with score of -1.

Qwen3-VL-32B-Thinking as the backbone model for online Reinforcement Learning (RL). To facilitate interaction during the RL rollout phase, we deployed a local instance of the WebArena environment on Alibaba Cloud ECS. Following the experimental setup of prior work Shen et al. (2025a), we utilize queries from the test set for training; however, no ground-truth annotations are accessed during this process. Since our reward function is entirely **ground-truth-agnostic**, this approach can be characterized as a Test-Time Training (TTT) strategy. Notably, unlike existing methods Shen et al. (2025a) that often rely on hand-crafted *tips* or site-specific instructions to guide the model, our training prompts are devoid of any domain-specific heuristics tailored to WebArena. This intentionally increases the complexity of the task and serves to underscore the intrinsic efficacy of our RL framework, demonstrating that the performance gains stem from the model's self-improvement rather than prompt engineering. The overall training reward is illustrated in Figure 8, which exhibits a consistent and healthy upward trend. For final evaluation, we report the performance at the 80th training step. During the inference phase, we maintain a rollout repetition of 5, *i.e.*, Pass@5, consistent with the training configuration. The final results are summarized in Table 5. As shown, our method achieves a 12.0% absolute improvement over TTI—which employs a similar TTT strategy—and an 10.7% gain over the vanilla Qwen3-VL-Thinking baseline.

Furthermore, we compare the performance of the vanilla Qwen3-VL-Thinking model and its RL-enhanced counterpart across different *Pass@K* metrics, as illustrated in Figure 9. At *Pass*@1, our RL-trained model outperforms the baseline by 8.08%. Notably, as K increases to 5, the performance gap widens to 10.66% (*Pass*@5). This increasing margin further demonstrates that the reinforcement learning process significantly reinforces the robustness of the model's predictions, enabling the agent to consistently identify successful execution paths within the expanded search space.

Table 5: Single model performance comparison across different web sites on WebArena. Our method demonstrates superior capabilities under the Test-Time Training setting.

Method	Backbone	Overall	Shopping	CMS	Reddit	GitLab	Maps
NNetnav Murty et al. (2024)	Llama 3.1 8B	7.2	7.4	4.2	0	0	28.5
AutoWebGLM Lai et al. (2024)	ChatGLM3 6B	18.2	-	-	-	-	-
AgentTrek Xu et al. (2025b)	Qwen2.5 32B	22.4	-	-	-	-	-
Learn-by-Interact Su et al.	Codestral 22B	24.2	-	-	-	-	-
TTI Shen et al. (2025a)	Gemma 3 12B	26.1	33.9	15.5	35.3	15.7	40.5
Baseline	Qwen3-VL-32B-Thinking	27.4	30.7	34.1	28.3	30.0	8.2
RL-HybridReward-Zero	Qwen3-VL-32B-Thinking	**38.1**	**40.0**	**37.6**	**36.1**	**35.4**	**42.3**

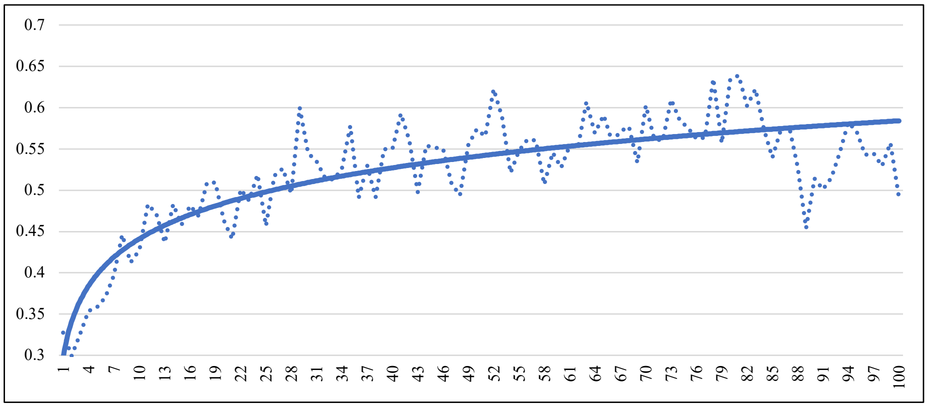

Figure 8: The training reward of Qwen3-VL-32B-Thinking-RL-HybridReward-Zero on WebArena.

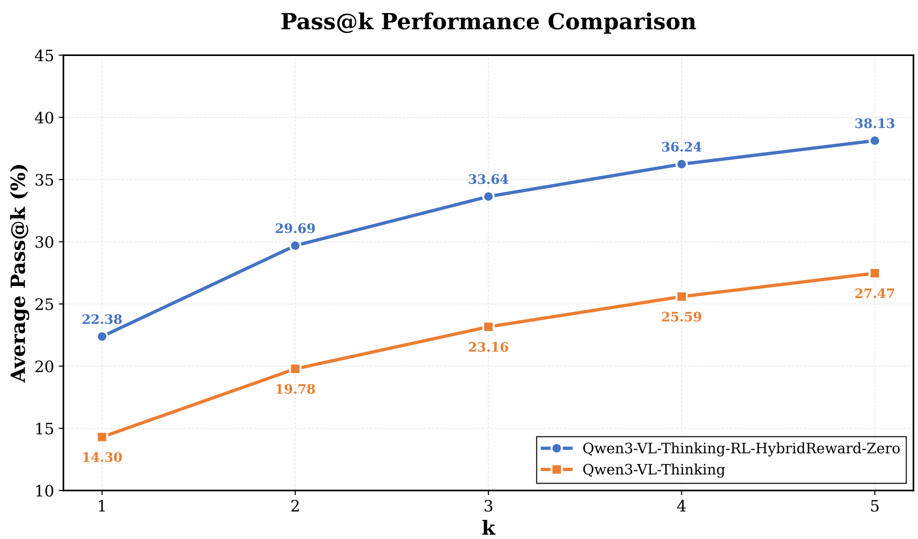

Figure 9: Performance comparison across different Pass@K on WebArena. We compare the base Qwen3-VL-32B-Thinking against the RL-tuned Qwen3-VL-32B-Thinking-RL-HybridReward-Zero.

4.3 OpAgent Architecture for Web

As previously demonstrated, relying solely on a standalone post-trained model is insufficient to achieve satisfactory performance in complex web agent tasks. To address this, we propose an agentic framework comprising four specialized components: a *Planner*, a *Grounder*, a *Reflector*, and a *Summarizer*. Specifically, we employ our supervised fine-tuned (SFT) Qwen2.5-VL-72B-MFT as the *Grounder*. For the remaining sub-modules evaluated on WebArena, we utilize Gemini-3-Pro as the underlying backend. Consistent with prior studies Zhou et al. (2026b); Marreed et al. (2025), we adopted a Human-in-the-Loop (HITL) strategy to incorporate site-specific tips for WebArena. Specifically, a tailored set of guidelines is curated for each individual website to provide the agent with necessary contextual hints. Ultimately, through a combination of refined prompt engineering and the seamless collaboration among our specialized agents, our framework attained the top position on the WebArena leaderboard, achieving a record-breaking score of 71.6%, as shown in Table 6.

Table 6: Agentic Architecture performance comparison across different web sites on WebArena.

Method	Backbone	Overall	Shopping	CMS	Reddit	GitLab	Maps
SteP Sodhi et al. (2023)	-	33.5	-	-	-	-	-
Agent Workflow Memory Wang et al. (2025c)	gpt-4	35.5	30.8	29.1	50.9	31.8	43.3
GUI-API Hybrid Agent Song et al. (2025)	gpt-4o	35.8	34.6	26.4	50.9	36.7	46.8
WebPilot Zhang et al. (2025b)	gpt-4o	37.2	36.9	24.7	65.1	39.4	33.9
AgentOccam-Judge Yang et al. (2025b)	gpt-4-turbo	45.7	46.2	38.9	67.0	43.3	52.3
Learn-by-Interact Su et al. (2025)	Codestral 22B	48.0	-	-	-	-	-
AgentSymbiotic Zhang et al. (2025a)	claude-3.5-sonnet	52.1	48.0	49.0	66.0	51.0	60.0
ScribeAgent Shen et al. (2024)	gpt-4o	53.0	45.8	37.9	73.7	59.7	56.3
WebOperator Dihan et al. (2025)	gpt-4o	54.6	49.2	55.0	76.4	52.8	55.2
Jace.AI	-	57.1	-	-	-	-	-
OpenAI Operator	-	58.1	-	-	-	-	-
IBM CUGA Marreed et al. (2025)	-	61.7	58.3	62.6	75.5	61.7	64.2
Narada AI	-	64.2	57.2	63.2	74.5	73.9	58.7
DeepSky Agent	-	66.9	-	-	-	-	-
GBOX AI	Claude Code	68.0	-	-	-	-	-
ColorBrowserAgent Zhou et al. (2026b)	GPT-5	71.2	**72.9**	**76.4**	**87.4**	65.7	55.9
OpAgent (Ours)	Gemini-3-Pro+Qwen2.5VL-MFT	**71.6**	59.2	71.3	86.0	**75.9**	**71.4**

5 Conclusion

In this work, we focused on the task of Web Automation through a series of integrated research efforts, including multi-task supervised fine-tuning (SFT), online Agentic Reinforcement Learning (RL) in the wild, and the construction of a modular OpAgent Architecture. By leveraging the collaborative synergy of multiple specialized agents, we achieved a record-breaking success rate of 71.6% on the WebArena benchmark, securing the top position on the leaderboard. However, the current paradigm remains heavily dependent on extensive prompt engineering and the complex orchestration of multiple agents, which incurs substantial human labor and computational overhead. Consequently, our future research will be directed toward enhancing the intrinsic exploration capabilities of individual models to reduce this reliance. Achieving this goal remains a formidable challenge.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025a. doi: 10.48550/ARXIV.2511.21631. URL `https://doi.org/10.48550/arXiv.2511.21631`.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan 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. *CoRR*, abs/2502.13923, 2025b. doi: 10.48550/ARXIV.2502.13923. URL `https://doi.org/10.48550/arXiv.2502.13923`.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse: From general vision language model to versatile GUI agent. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria,* *July 27 - August 1, 2025*, pp. 21936–21959. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1065/.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025. doi: 10.48550/ARXIV.2505.22617. URL https://doi.org/10.48550/arXiv.2505.22617.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html.

Mahir Labib Dihan, Tanzima Hashem, Mohammed Eunus Ali, and Md. Rizwan Parvez. Weboperator: Action-aware tree search for autonomous agents in web environment. *CoRR*, abs/2512.12692, 2025. doi: 10.48550/ARXIV.2512.12692. URL https://doi.org/10.48550/arXiv.2512.12692.

Gonzalo Gonzalez-Pumariega, Vincent Tu, Chih-Lun Lee, Jiachen Yang, Ang Li, and Xin Eric Wang. The unreasonable effectiveness of scaling agents for computer use. *CoRR*, abs/2510.02250, 2025. doi: 10.48550/ARXIV.2510.02250. URL https://doi.org/10.48550/arXiv.2510.02250.

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 *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=kxnoqaisCT.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=c6l7yA0HSq.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, volume 11220 of *Lecture Notes in Computer Science*, pp. 282–299. Springer, 2018. doi: 10.1007/978-3-030-01270-0_17. URL https://doi.org/10.1007/978-3-030-01270-0_17.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 6864–6890. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.371. URL https://doi.org/10.18653/v1/2024.acl-long.371.

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

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. Autowebglm: A large language model-based web navigating agent. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 5295–5306. ACM, 2024. doi: 10.1145/3637528.3671620. URL https://doi.org/10.1145/3637528.3671620.

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. *CoRR*, abs/2504.07981, 2025. doi: 10.48550/ARXIV.2504.07981. URL https://doi.org/10.48550/arXiv.2504.07981.

Bingchang Liu, Chaoyu Chen, Zi Gong, Cong Liao, Huan Wang, Zhichao Lei, Ming Liang, Dajun Chen, Min Shen, Hailian Zhou, et al. Mftcoder: Boosting code llms with multitask fine-tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5430–5441, 2024.

Haowei Liu, Xi Zhang, Haiyang Xu, Yuyang Wanyan, Junyang Wang, Ming Yan, Ji Zhang, Chunfeng Yuan, Changsheng Xu, Weiming Hu, and Fei Huang. Pc-agent: A hierarchical multi-agent collaboration framework for complex task automation on PC. *CoRR*, abs/2502.14282, 2025. doi: 10.48550/ARXIV.2502.14282. URL https://doi.org/10.48550/arXiv.2502.14282.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. GUI-R1 : A generalist r1-style vision-language action model for GUI agents. *CoRR*, abs/2504.10458, 2025. doi: 10.48550/ARXIV.2504.10458. URL https://doi.org/10.48550/arXiv.2504.10458.

Sami Marreed, Alon Oved, Avi Yaeli, Segev Shlomov, Ido Levy, Aviad Sela, Asaf Adi, and Nir Mashkif. Towards enterprise-ready computer using generalist agent. *CoRR*, abs/2503.01861, 2025. doi: 10.48550/ARXIV.2503.01861. URL https://doi.org/10.48550/arXiv.2503.01861.

Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D Manning. Nnetnav: Unsupervised learning of browser agents through environment interaction in the wild. *arXiv preprint arXiv:2410.02907*, 2024.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Junhong Shen, Atishay Jain, Zedian Xiao, Ishan Amlekar, Mouad Hadji, Aaron Podolny, and Ameet Talwalkar. Scribeagent: Towards specialized web agents using production-scale workflow data. *CoRR*, abs/2411.15004, 2024. doi: 10.48550/ARXIV.2411.15004. URL https://doi.org/10.48550/arXiv.2411.15004.

Junhong Shen, Hao Bai, Lunjun Zhang, Yifei Zhou, Amrith Setlur, Shengbang Tong, Diego Caples, Nan Jiang, Tong Zhang, Ameet Talwalkar, and Aviral Kumar. Thinking vs. doing: Agents that reason by scaling test-time interaction. *CoRR*, abs/2506.07976, 2025a. doi: 10.48550/ARXIV.2506.07976. URL https://doi.org/10.48550/arXiv.2506.07976.

Junhong Shen, Hao Bai, Lunjun Zhang, Yifei Zhou, Amrith Setlur, Shengbang Tong, Diego Caples, Nan Jiang, Tong Zhang, Ameet Talwalkar, et al. Thinking vs. doing: Agents that reason by scaling test-time interaction. *arXiv preprint arXiv:2506.07976*, 2025b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Paloma Sodhi, SRK Branavan, Yoav Artzi, and Ryan McDonald. Step: Stacked llm policies for web actions. *arXiv preprint arXiv:2310.03720*, 2023.

Yueqi Song, Frank F. Xu, Shuyan Zhou, and Graham Neubig. Beyond browsing: Api-based web agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 11066–11085. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.577/.

Hongjin Su, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan O Arik. Learn-byinteract: A data-centric framework for self-adaptive agents in realistic environments. corr, abs/2501.10893, 2025. doi: 10.48550. *arXiv preprint ARXIV.2501.10893*.

Hongjin Su, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan Ö. Arik. Learn-by-interact: A data-centric framework for self-adaptive agents in realistic environments. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=3UKOzGWCVY.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. doi: 10.48550/ARXIV.2507.06261. URL https://doi.org/10.48550/arXiv.2507.06261.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025a.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Keqin Chen, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/dc06d4d2792265fb5454a6092bfd5c6a-Abstract-Conference.html.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, JingJing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025b. doi: 10.48550/ARXIV.2508.18265. URL https://doi.org/10.48550/arXiv.2508.18265.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025c. URL https://openreview.net/forum?id=NTAhi2JEEE.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *CoRR*, abs/2505.16421, 2025. doi: 10.48550/ARXIV.2505.16421. URL https://doi.org/10.48550/arXiv.2505.16421.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. Gui-actor: Coordinate-free visual grounding for gui agents. *arXiv preprint arXiv:2506.03143*, 2025.

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

Yifan Xu, Xiao Liu, Xinghan Liu, Jiaqi Fu, Hanchen Zhang, Bohao Jing, Shudan Zhang, Yuting Wang, Wenyi Zhao, and Yuxiao Dong. Mobilerl: Online agentic reinforcement learning for mobile GUI agents. *CoRR*, abs/2509.18119, 2025a. doi: 10.48550/ARXIV.2509.18119. URL https://doi.org/10.48550/arXiv.2509.18119.

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. *arXiv preprint arXiv:2412.04454*, 2024.

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=EEgYUccwsV.

Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? assessing the current state of web agents. *CoRR*, abs/2504.01382, 2025. doi: 10.48550/ARXIV.2504.01382. URL https://doi.org/10.48550/arXiv.2504.01382.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025a. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=oWdzUpOlkX.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, Jitong Liao, Qi Zheng, Fei Huang, Jingren Zhou, and Ming Yan. Mobile-agent-v3: Fundamental agents for GUI automation. *CoRR*, abs/2508.15144, 2025. doi: 10.48550/ARXIV.2508.15144. URL https://doi.org/10.48550/arXiv.2508.15144.

Ruichen Zhang, Mufan Qiu, Zhen Tan, Mohan Zhang, Vincent Lu, Jie Peng, Kaidi Xu, Leandro Z. Agudelo, Peter Qian, and Tianlong Chen. Symbiotic cooperation for web agents: Harnessing complementary strengths of large and small llms. *CoRR*, abs/2502.07942, 2025a. doi: 10.48550/ARXIV.2502.07942. URL https://doi.org/10.48550/arXiv.2502.07942.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 23378–23386. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I22.34505. URL https://doi.org/10.1609/aaai.v39i22.34505.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=piecKJ2DlB.

Jiamu Zhou, Jihong Wang, Weiming Zhang, Weiwen Liu, Zhuosheng Zhang, Xingyu Lou, Weinan Zhang, Huarong Deng, and Jun Wang. Colorbrowseragent: An intelligent gui agent for complex long-horizon web automation. *arXiv preprint arXiv:2601.07262*, 2026a.

Jiamu Zhou, Jihong Wang, Weiming Zhang, Weiwen Liu, Zhuosheng Zhang, Xingyu Lou, Weinan Zhang, Huarong Deng, and Jun Wang. Colorbrowseragent: An intelligent gui agent for complex long-horizon web automation. *arXiv preprint arXiv:2601.07262*, 2026b.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. GUI-G1: understanding r1-zero-like training for visual grounding in GUI agents. *CoRR*, abs/2505.15810, 2025. doi: 10.48550/ARXIV.2505.15810. URL https://doi.org/10.48550/arXiv.2505.15810.