UI-R1: Enhancing Efficient Action Prediction of GUI Agents by Reinforcement Learning

Zhengxi Lu[1,2*], Yuxiang Chai[3*], Yaxuan Guo[2], Xi Yin[2], Liang Liu[2†], Hao Wang[2], Han Xiao[3], Shuai Ren[2], Pengxiang Zhao[1], Guangyi Liu[1], Guanjing Xiong[2], Hongsheng Li[3‡]

[1] Zhejiang University
[2] vivo AI Lab
[3] MMLab @ CUHK
zhengxilu@zju.edu.cn, hsli@ee.cuhk.edu.hk

Abstract

The recent DeepSeek-R1 has showcased the emergence of reasoning capabilities in LLMs through reinforcement learning (RL) with rule-based rewards. Despite its success in language models, its application in multimodal domains, particularly in graphic user interface (GUI) agent tasks, remains under-explored. To address this issue, we propose **UI-R1**, the first framework to explore how rule-based RL can enhance the reasoning capabilities of multimodal large language models (MLLMs) for GUI action prediction tasks. UI-R1 introduces a novel rule-based action reward scheme, facilitating model optimization via policy-based algorithms such as Group Relative Policy Optimization (GRPO). To further improve efficiency during inference, we present **UI-R1-E**fficient, a two-stage training paradigm that both shortens reasoning length and enhances overall performance. Additionally, we construct a compact yet high-quality dataset comprising 2K challenging tasks across five prevalent mobile device action types. Experimental results show that our proposed models (e.g., UI-R1-3B) achieve substantial improvements over the base model (i.e., Qwen2.5-VL-3B) on both in-domain (ID) and out-of-domain (OOD) tasks, with average accuracy gains of **18.3%** on ScreenSpot, **6.0%** on ScreenSpot-Pro, and **10.9%** on ANDROIDCONTROL. Moreover, our efficient versions deliver competitive performance compared to considerably larger state-of-the-art models. These results underscore the potential of reinforcement learning to advance GUI control, paving the way for future research in Human-Computer Interaction (HCI).

Code — https://github.com/lll6gg/UI-R1
Datasets —
https://huggingface.co/datasets/LZXzju/UI-R1-3B-Train
Extended version — https://arxiv.org/abs/2503.21620

Introduction

Supervised fine-tuning (SFT) has long been the standard training paradigm for large language models (LLMs) and graphic user interface (GUI) agents (Qin et al. 2025; Wu et al. 2024; Hong et al. 2024; Tang et al. 2025). However, SFT relies heavily on large-scale, high-quality labeled datasets, leading to prolonged training times and high computational costs. Furthermore, existing open-source VLM-based GUI agents trained using SFT can be criticized for poor performance in out-of-domain (OOD) scenarios (Lu et al. 2024; Chai et al. 2024), limiting their effectiveness and applicability in real-world applications.

Rule-based reinforcement learning or reinforcement fine-tuning (RFT) has recently emerged as an efficient and scalable alternative to SFT for the development of LLMs, which efficiently fine-tune the model with merely dozens to thousands of samples to excel at domain-specific tasks. It uses predefined task-specific reward functions, eliminating the need for costly human annotations. Recent works, such as DeepSeek-R1 (Guo et al. 2025), demonstrate the effectiveness of rule-based RL in mathematical problem solving by evaluating the correctness of the solution, while others (Liu et al. 2025b; Wang et al. 2025b; Peng et al. 2025; Chen et al. 2025; Huang et al. 2025; Zhou et al. 2025; Chen, Luo, and Li 2025) extend the algorithm to multimodal models, achieving notable improvements in vision-related tasks such as image grounding and object detection. By focusing on measurable objectives, rule-based RL enables practical and versatile model optimization across both textual and multimodal domains, offering significant advantages in terms of efficiency, scalability, and reduced reliance on large datasets.

However, existing related studies always target on general vision-related tasks like grounding and detection using Intersection over Union (IoU) metric. In this work, we extend the rule-based RL paradigm to a new application domain by focusing on GUI action prediction tasks driven by low-level instructions. Our proposed reward function evaluates each generated response and updates the model by policy optimization, such as GRPO (Shao et al. 2024). In detail, our action-based reward function contains the action type reward, the action argument reward, along with the commonly used format reward. This flexible and effective reward mechanism is well aligned with the objectives of general GUI-related tasks, enhancing model's reasoning capabilities of action prediction by iterative self-learning.

[*] These authors contributed equally.
[†] Project Lead
[‡] Corresponding Author

Regarding data preparation, we follow Muennighoff et al. (2025) and select just 2K training samples according to three criterion: difficulty, diversity, and quality, making our method remarkably data-efficient. Experiments demonstrate that UI-R1 achieves significant performance improvements on out-of-domain gounding tasks like ScreenSpot-Pro (Li et al. 2025a) and computer scenarios in ScreenSpot (Cheng et al. 2024), indicating the potential of rule-based RL to tackle complex GUI-related tasks across diverse domains effectively. Furthermore, our novel UI-R1-E framework enables efficient reasoning for GUI grounding via two-stage length reward function, demonstrating that *for certain visual tasks, less extensive reasoning can actually lead to better performance*. This finding challenges the conventional belief that more reasoning is always better and provides new insights for designing efficient MLLM-based GUI agents, advancing human-computer interaction (HCI).

In summary, our contributions are as follows.

- We propose UI-R1, the **first** framework which enhances MLLM's reasoning capabilities on GUI action prediction tasks through DeepSeek R1 style reinforcement learning.
- We design a rule-based action reward function that effectively aligns with the objectives of common GUI tasks, facilitating the self-refinement and iterative optimization of the policy model.
- We adopt a three-stage data selection strategy to curate high-quality training datasets (i.e., UI-R1-0.1K and UI-R1-2K). Remarkably, our models **UI-R1-3B** and **UI-R1-7B**, trained on the minimal UI-R1-0.1K set containing only 136 samples, achieve substantial performance improvements on out-of-domain benchmarks, including challenging desktop and web scenarios.
- Our enhanced variants, **UI-R1-E-3B** and **UI-R1-E-7B**, trained on the broader UI-R1-2K set, yield significant improvements in both grounding efficiency and prediction accuracy over the base UI-R1 models.

Related Work

GUI Agents

Starting with CogAgent (Hong et al. 2024), researchers have used MLLMs for GUI-related tasks, including device control, task completion, GUI understanding, and more (Liu et al. 2025a). One line of work, such as the AppAgent series (Zhang et al. 2023; Li et al. 2024b) and the Mobile-Agent series (Wang et al. 2024b,a), integrates commercial generalist models like GPT for planning and prediction tasks. These agents rely heavily on prompt engineering and multi-agent collaboration to execute complex tasks, making them adaptable but dependent on careful manual design for optimal performance. Another branch of research focuses on fine-tuning smaller open-source MLLMs on task-specific GUI datasets (Rawles et al. 2023; Li et al. 2024a; Chai et al. 2024; Gou et al. 2024) to create specialist agents. For example, Chai et al. (2024) enhances agents by incorporating additional functionalities of the GUI element in the Android system, while UGround(Gou et al. 2024) develops a special GUI grounding model tailored for precise localization of the GUI element. Wu et al. (2024) develops a foundational model for GUI action prediction. Moving beyond task-specific fine-tuning, UI-TARs (Qin et al. 2025) introduces a more comprehensive approach by combining GUI-related pretraining with task-wise reasoning fine-tuning, aiming to better align models with the intricacies of GUI interactions. Despite their differences, all of these existing agents share a common reliance on the SFT paradigm, which depends heavily on large-scale, high-quality labeled datasets.

Rule-Based Reinforcement Learning

Rule-based reinforcement learning has recently emerged as an efficient alternative to traditional training paradigms by leveraging predefined rule-based reward functions to guide model behavior. DeepSeek-R1 (Guo et al. 2025) first introduced this approach, using reward functions based on predefined criteria, such as checking whether an LLM's final answer matches the ground truth for math problems. The reward focuses solely on the final results, leaving the reasoning process to be learned by the model itself. Zeng et al. (2025) reproduces the algorithm on models with smaller sizes and illustrates its effectiveness on small language models. Subsequent works (Chen et al. 2025; Shen et al. 2025a; Liu et al. 2025b; Wang et al. 2025b; Peng et al. 2025; Meng et al. 2025), extended the paradigm to multimodal models by designing task-specific rewards for visual tasks, including correct class predictions for image classification and IoU metrics for image grounding and detection. These studies demonstrate the adaptability of rule-based RL for both pure-language and multimodal models. By focusing on task-specific objectives without requiring extensive labeled datasets or human feedback, rule-based RL shows strong potential as a scalable and effective training paradigm across diverse tasks.

Efficient Reasoning. Recent studies (Xiao et al. 2025; Shen et al. 2025b; Aggarwal and Welleck 2025; Li et al. 2025b; Wang et al. 2025a) have introduced a **length reward** within reinforcement learning frameworks, which encourages concise and accurate reasoning by rewarding short, correct answers and penalizing lengthy or incorrect ones.

Method

UI-R1 is a RL training paradigm designed to enhance a GUI agent's ability under *low-level instructions* [1]. Beyond that, we develop UI-R1-E framework to reason efficiently and perform accurately via two-stage RL.

Preliminary

Many rule-based RL works (Guo et al. 2025; Zeng et al. 2025; Liu et al. 2025b) adopt the Group Relative Policy Optimization (GRPO) algorithm (Shao et al. 2024) for RL

[1] We define **low-level instructions** as directives that guide the agent to perform actions based on a single state (e.g., a GUI screenshot), consistent with the definition in ANDROIDCONTROL (Li et al. 2024a). For example, *"Click the menu icon in the top left corner"* represents a low-level instruction, whereas *"Create an event for 2 PM tomorrow"* is a high-level instruction.

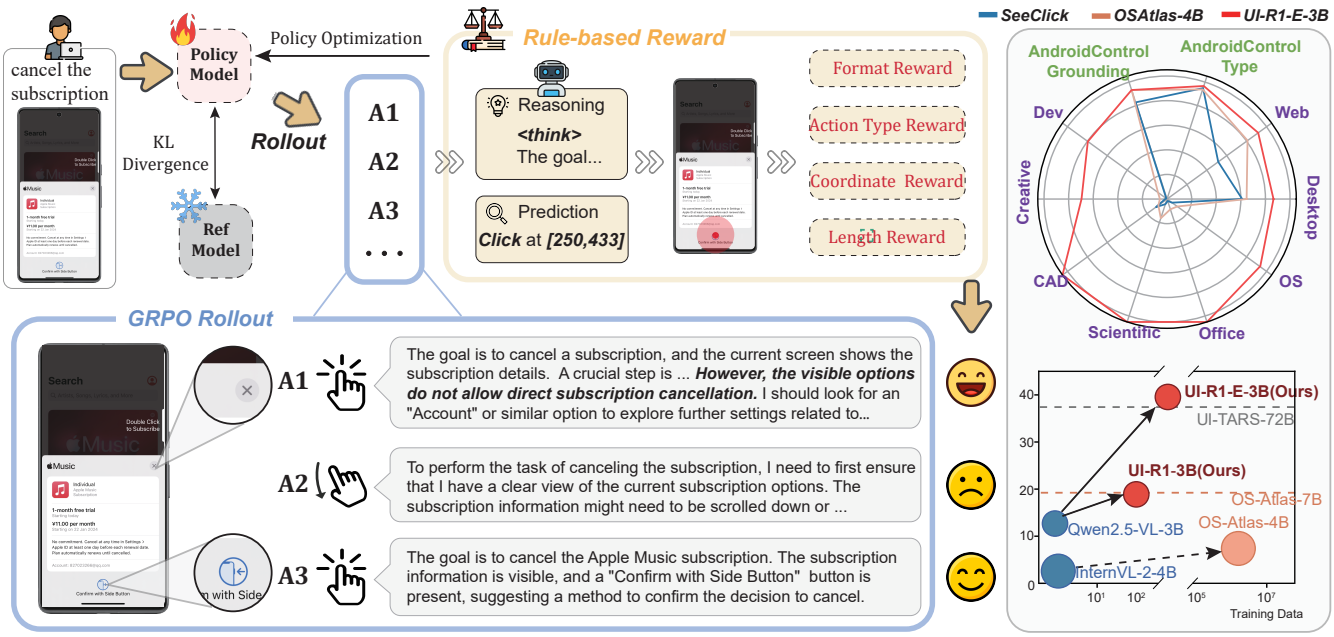

Figure 1: Overview of UI-R1 training framework and 3B model's performance. **Left**: Given a GUI screenshot and a text instruction from the user, the policy model (i.e., Qwen2.5-VL-3B) generates multiple action planning responses with reasoning. Our proposed rule-based action reward function is then applied, and the policy model is updated using a policy gradient optimization algorithm. **Right top**: Our UI-R1-E-3B achieves SOTA on both in-domain (i.e., AndroidControl) and out-of-domain (i.e., ScreenSpot-Pro, ScreenSpot desktop and web subsets) tasks (UI-TARS-72B as 100%). **Right bottom**: Employing reinforcement fine-tuning (RFT), UI-R1-3B and UI-R1-E-3B achieve performance comparable to SFT models (i.e., OS-Atlas-7B and UI-TARA-72B) on ScreenSpot-Pro with significantly fewer data and smaller model size (indicated by the circle radius).

training. GRPO offers an alternative to commonly used Proximal Policy Optimization (PPO) (Schulman et al. 2017) by eliminating the need for a critic model. Instead, GRPO directly compares a group of candidate responses to determine their relative quality. Given a task question, the model generates a set of N potential responses $\{o_1, o_2, \ldots, o_N\}$. Each response is evaluated by taking the corresponding actions and computing its reward $\{r_1, r_2, \ldots, r_N\}$. The policy model is optimized by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = E_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \left\{ \min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \quad \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon \right)\hat{A}_{i,t} \right] - \beta D_{KL}\left[\pi_\theta||\pi_{ref}\right] \right\} \right] \tag{1}$$

where π_θ and π_{old} are the current and old policy, and ϵ and β are hyper-parameters introduced in PPO.

The relative quality A_i of the i-th response is computed as

$$A_i = \frac{r_i - Mean(\{r_1, r_2, \ldots, r_N\})}{Std(\{r_1, r_2, \ldots, r_N\})}, \tag{2}$$

where $Mean$ and Std represent the mean and standard deviation of the rewards, respectively. This normalization step ensures that responses are compared within the context of the group, allowing GRPO to better capture nuanced differences between candidates. Unlike PPO, which relies on a single reward signal and a critic to estimate the value function, GRPO normalizes these rewards to calculate the relative advantage of each response.

Rule-Based Action Rewards

The rule-based reward function introduced by DeepSeek-R1 (Guo et al. 2025) represents a foundational step in rule-based RL by simply evaluating whether model predictions exactly match ground-truth answers. This straightforward approach efficiently aligns models with preference alignment algorithms and provides clear optimization signals. For vision-related tasks, works such as VLM-R1 (Shen et al. 2025a) and Visual-RFT (Liu et al. 2025b) extend this idea by designing task-specific rewards. For image grounding tasks, they compute the IoU between the predicted and ground-truth bounding boxes as the reward. Similarly, for image classification tasks, rewards are determined by checking whether the predicted and ground-truth classes match.

In GUI-related tasks, grounding is a critical requirement for agents. Unlike traditional image grounding tasks, GUI grounding requires agents to identify where specific actions, such as `click`, should be performed on a given GUI screen-

shot. To address this unique gap, we propose a reward function tailored for GUI tasks, as defined in Equation 3:

$$R = R_{\mathcal{T}} + R_{\mathcal{C}} + R_{\mathcal{F}}, \tag{3}$$

where the predicted action $\mathcal{A} = \{\mathcal{T}, \mathcal{C}\}$ consists of two components: $\mathcal{T}$, which represents the action type (e.g., `click`, `scroll`), and $\mathcal{C}$, which represents the `click` coordinate. $R_{\mathcal{F}}$ represents the commonly used response format reward.

Action type reward. In our tasks, the action space includes `Click`, `Scroll`, `Back`, `Open_App`, and `Input_Text`, covering a wide range of common application scenarios in daily life, as inspired by GUIPivot (Wu et al. 2025). The action type reward, denoted as $\boldsymbol{R_{\mathcal{T}}}$, is computed by comparing the predicted action type $\mathcal{T}'$ with the ground truth action type $\mathcal{T}$. It assigns a reward of 1 if $\mathcal{T}' = \mathcal{T}$ and 0 otherwise, providing a straightforward and effective evaluation mechanism for action type prediction.

Coordinate accuracy reward. Through observation, we find that among all action types, the most common action argument error occurs in the mis-prediction of coordinates for the `click` action when given a low-level instruction. To address this issue, we specifically design a coordinate accuracy reward. The model is required to output a coordinate $\mathcal{C} = [x, y]$, indicating where the `click` action should be performed. Given the ground truth bounding box $\mathcal{B} = [x1, y1, x2, y2]$, the coordinate accuracy reward $R_{\mathcal{C}}$ is computed as shown in Equation 4:

$$R_{\mathcal{C}} = \begin{cases} 1 & \text{if coord } \mathcal{C} \textit{ in } \text{box } \mathcal{B}, \\ 0 & \text{else.} \end{cases} \tag{4}$$

Unlike general visual grounding tasks which compute the IoU between the predicted bounding box and the ground truth box, our approach prioritizes action coordinate prediction over element grounding. This focus is more appropriate for GUI agents and better aligns with human intuition, as the ultimate goal is to ensure correct actions are performed rather than merely locating GUI elements.

Format reward. During training, we incorporate the widely-used format reward to guide the model in generating its reasoning process and final answer in a structured format. The format reward, denoted as $\boldsymbol{R_{\mathcal{F}}}$, ensures that the model's predictions follow the required HTML tag format, specifically using $<\boldsymbol{think}>$ for the reasoning process and $<\boldsymbol{answer}>$ for the final answer. This structured output not only enhances clarity, but also ensures consistency in the model's predictions.

Efficient reasoning

Our optimized grounding version, **UI-R1-E**fficient, is trained in two stages: DAST training followed by NOTHINK training, with each stage lasting 4 epochs.

DAST refines rule-based rewards by integrating the deviation between actual response length and the Token Length Budget (TLB) metric, allowing the reward to capture both task difficulty and length properties for effective difficulty-adaptive training (Shen et al. 2025b).

TLB metric is defined as:

$$L_{budget} = p \cdot L_{\overline{r}} + (1 - p) \cdot L_{max}, p = \frac{c}{N} \tag{5}$$

where c is the number of correct responses sampled for a given question, and N is the total number of sampled responses. $L_{\overline{r}}$ denotes the average token length of the correct responses, and $L_{\max}$ represents the maximum generation length.

The length reward $R_{\mathcal{L}}$ is defined as:

$$R_{\mathcal{L}} = \begin{cases} \max(-0.5\lambda + 0.5,\ 0.1) & \textit{if correct} \\ \min(0.9\lambda - 0.1,\ -0.1) & \textit{if incorrect} \end{cases} \tag{6}$$

where $\lambda = \dfrac{L_i - L_{budget}}{L_{budget}}$. The reward in Equation 3 is then calibrated by the length reward:

$$R = R_{\mathcal{T}} + R_{\mathcal{C}} + R_{\mathcal{F}} + R_{\mathcal{L}}, \tag{7}$$

NOTHINK enables fast and direct grounding by removing the $<\boldsymbol{think}>$ tags, thereby bypassing explicit reasoning steps during both training and inference (Li et al. 2025b).

Training Data Selection

Compared to SFT, rule-based RL has demonstrated the capability to achieve comparable or even superior performance on mathematical and vision-related tasks using only a limited number of training samples (Zeng et al. 2025; Liu et al. 2025b). Inspired by s1 (Muennighoff et al. 2025) and GUI-R1 (Luo et al. 2025), we implement a three-stage data selection process to refine open-source GUI-related datasets based on three key principles: Quality, Difficulty, and Diversity.

Quality. For refining the `click` action arguments, we collect data related to GUI grounding tasks from Amex (Chai et al. 2024) and FineWeb (Penedo et al. 2024). For other actions, we randomly select 1K episodes from ANDROIDCONTROL (Li et al. 2024a).

Difficulty. To identify hard samples, we evaluated Qwen2.5-VL-3B (Bai et al. 2025) on each task instruction by model performance, where a sample is labeled "hard" if the model's output does not match the ground truth. We only keep the "hard" samples among all the data collected.

Diversity. To ensure data diversity, we select samples featuring a range of action types in ANDROIDCONTROL (e.g., `Scroll`, `Back`, `Open App`, `Input Text`) and element types in Amex and FineWeb (e.g., `Icon`, `Text`). Rare or less informative actions, such as `Wait` and `Long Press`, are excluded from ANDROIDCONTROL.

Based on these criteria, we construct a high-quality training dataset comprising 2,000 samples, referred to as **UI-R1-2K**. Our original smaller dataset, which is consisted of 136 samples and also curated using this method, is denoted as **UI-R1-0.1K**.

Model	Development		Creative		CAD		Scientific		Office		OS		Avg
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Qwen-VL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.1
GPT-4o	1.3	0.0	1.0	0.0	2.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	0.8
SeeClick-9.6B	0.6	0.0	1.0	0.0	2.5	0.0	3.5	0.0	1.1	0.0	2.8	0.0	1.1
Qwen2-VL-7B	2.6	0.0	1.5	0.0	0.5	0.0	6.3	0.0	3.4	1.9	0.9	0.0	1.6
ShowUI-2B	16.9	1.4	9.1	0.0	2.5	0.0	13.2	7.3	15.3	7.5	10.3	2.2	7.7
OS-Atlas-4B	7.1	0.0	3.0	1.4	2.0	0.0	9.0	5.5	5.1	3.8	5.6	0.0	3.7
CogAgent-18B	14.9	0.7	9.6	0.0	7.1	3.1	22.2	1.8	13.0	0.0	5.6	0.0	7.7
Aria-UI	16.2	0.0	23.7	2.1	7.6	1.6	27.1	6.4	20.3	1.9	4.7	0.0	11.3
UGround-7B	26.6	2.1	27.3	2.8	14.2	1.6	31.9	2.7	31.6	11.3	17.8	0.0	16.5
Claude**	22.0	3.9	25.9	3.4	14.5	3.7	33.9	15.8	30.1	16.3	11.0	4.5	17.1
OS-Atlas-7B	33.1	1.4	28.8	2.8	12.2	4.7	37.5	7.3	33.9	5.7	27.1	4.5	18.9
Qwen2.5-VL-3B	14.9	2.1	20.2	1.4	4.1	4.7	34.0	7.3	22.0	3.8	6.5	2.2	11.8
Qwen2.5-VL-7B	33.1	2.1	23.7	3.5	12.2	6.3	36.8	7.3	37.8	7.5	30.8	6.9	19.3
Qwen2.5-VL-3B*	20.3	1.8	24.6	2.8	11.2	4.7	39.5	6.4	28.6	5.7	17.8	2.2	15.8
Qwen2.5-VL-7B*	31.4	1.8	27.3	3.5	15.7	5.1	40.7	7.9	39.7	8.9	32.4	6.9	20.7
GUI-R1-3B	33.8	4.8	40.9	5.6	26.4	7.8	61.8	17.3	53.6	17.0	28.1	5.6	28.6
GUI-R1-7B	49.4	4.8	38.9	8.4	23.9	6.3	55.6	11.8	58.7	26.4	42.1	**16.9**	31.3
UI-R1-3B (Ours)	22.7	4.1	27.3	3.5	11.2	6.3	42.4	11.8	32.2	11.3	13.1	4.5	17.8
UI-R1-7B (Ours)	42.9	4.1	43.9	**18.2**	21.3	**17.2**	25.0	12.7	27.1	**54.7**	39.3	11.2	26.4
UI-R1-E-3B (Ours)	46.1	6.9	41.9	4.2	37.1	12.5	56.9	**21.8**	**65.0**	26.4	32.7	10.1	33.5
UI-R1-E-7B (Ours)	**59.1**	**12.4**	**60.1**	11.2	**42.6**	12.5	**67.4**	21.8	38.4	50.9	**52.3**	12.4	**39.2**

Table 1: Grounding accuracy on ScreenSpot-Pro. The optimal and the suboptimal results are **bolded** and underlined, respectively. Claude** refers to *Claude-computer-use*.

Experiments

Models. We fine-tune Qwen2.5-VL-3B and Qwen2.5-VL-7B using our UI-R1 and UI-R1-E frameworks. Specifically, UI-R1 is trained on the compact UI-R1-0.1K dataset, while UI-R1-E is trained on the expanded UI-R1-2K dataset, yielding four models: **UI-R1-3B**, **UI-R1-7B**, **UI-R1-E-3B**, and **UI-R1-E-7B**.

Baselines. For comparison, we select current competitive models listed as follows.

- **Proprietary Models:** GPT-4o (OpenAI 2024) and Claude (Anthropic 2024).
- **General Open-source Models:** Qwen-VL-7B (Bai et al. 2023), Qwen2-VL-7B (Wang et al. 2024c), Qwen2.5-VL-3B (Bai et al. 2025), Qwen2.5-VL-7B (Bai et al. 2025).
- **GUI-specific Models:** CogAgent (Hong et al. 2024), SeeClick (Cheng et al. 2024), UGround (Gou et al. 2024), AGUVIS (Xu et al. 2024), Aria-UI (Yang et al. 2024), OS-Atlas (Wu et al. 2024). To demonstrate UI-R1's generalizability, we train the base model using supervised fine-tuning on the same 2K samples, referring as Qwen2.5-VL-3B* and Qwen2.5-VL-7B* in Table 1, Table 2 and Table 3.

GUI Grounding Capability

Grounding capability is evaluated on two benchmarks: ScreenSpot (Cheng et al. 2024) and ScreenSpot-Pro (Li et al. 2025a). ScreenSpot evaluates GUI grounding capability across mobile, desktop, and web platforms, while ScreenSpot-Pro focuses on high-resolution professional environments, featuring expert-annotated tasks spanning 23 applications, five industries, and three operating systems.

Metric. We evaluate accuracy as the proportion of test samples where the predicted `click` coordinate falls within the ground truth bounding box.

Model	Mobile		Web		Desktop		Avg.
	Icon	Text	Icon	Text	Icon	Text	
CogAgent-18B	24.0	67.0	28.6	70.4	20.0	74.2	47.4
SeeClick-9.6B	52.0	78.0	32.5	55.7	30.0	72.2	53.4
Qwen2-VL-7B	60.7	75.5	25.7	35.2	54.3	76.3	55.3
UGround-V1-7B	60.3	82.8	70.4	80.4	63.6	82.5	73.3
Qwen2.5-VL-3B	61.1	90.5	43.2	60.0	40.0	80.9	65.0
Qwen2.5-VL-7B	69.4	94.5	65.1	86.9	60.0	89.7	79.3
Qwen2.5-VL-3B*	71.2	95.2	63.1	78.3	46.4	85.0	75.7
Qwen2.5-VL-7B*	77.7	94.1	68.2	87.8	62.8	90.3	81.8
AGUVIS-7B	78.2	88.3	70.7	88.1	74.8	85.7	81.8
GUI-R1-3B	78.2	97.6	72.4	91.0	64.3	94.3	85.0
GUI-R1-7B	86.4	98.8	77.2	92.1	**79.4**	92.3	88.7
UI-R1-3B (Ours)	84.7	95.6	73.3	85.2	59.3	90.2	83.3
UI-R1-7B (Ours)	86.2	94.7	72.1	92.6	74.5	93.3	86.7
UI-R1-E-3B (Ours)	83.0	97.1	**85.0**	91.7	77.9	95.4	89.2
UI-R1-E-7B (Ours)	**89.6**	**99.2**	84.2	**93.6**	77.8	**95.8**	**91.2**

Table 2: Grounding accuracy on ScreenSpot. The best and second-best results are **bolded** and underlined, respectively.

Analysis. Experimental results demonstrate that our approach substantially enhances the GUI grounding performance of the base models, with improvements of up to **+24.2%** on ScreenSpot and **+19.9%** on ScreenSpot-Pro (as shown in Table 2 and Table 1). In particular, UI-R1-E-3B achieves scores of 89.2 on ScreenSpot and 33.5 on ScreenSpot-Pro, while UI-R1-E-7B reaches 91.2 and 39.2 respectively, establishing new state-of-the-art results among models of similar scale and methodology (e.g., GUI-R1-3B and GUI-R1-7B).

Qwen2.5-VL-3B* and Qwen2.5-VL-7B* in Table 2 demonstrate that supervised fine-tuning (SFT) with a limited amount of data (e.g., 2K samples) can effectively improve in-domain performance by tailoring the model to specific tasks. However, the comparison between Qwen2.5-VL-3B* and Qwen2.5-VL-3B in Table 1 highlights a critical limitation of SFT: its effectiveness significantly diminishes in OOD scenarios. This limitation arises from the dependency of SFT on task-specific labeled data, restricting the model's ability to adapt to unseen environments. In contrast, our RL approach not only enhances OOD generalization by focusing on task-specific reward optimization, but also achieves with far fewer training samples, offering a scalable and efficient alternative to traditional SFT methods.

Action Prediction Capability

We further evaluate the model's ability to predict single-step actions from low-level instructions on the ANDROIDCONTROL benchmark, which features a broader and more diverse action space.

Model	Type	GR	SR	Overall
GPT-4o	74.3	38.7	28.4	47.1
OS-Atlas-4B	64.6	71.2	67.9	67.9
OS-Atlas-7B	73.0	73.4	**73.2**	73.2
Qwen2.5-VL-3B	62.0	74.1	59.3	65.1
Qwen2.5-VL-7B	83.4	<u>87.0</u>	62.5	77.6
Qwen2.5-VL-3B*	80.5	79.4	67.8	75.9
Qwen2.5-VL-7B*	78.0	**87.1**	68.7	77.9
GUI-R1-3B	83.7	81.6	64.4	76.6
GUI-R1-7B	85.2	85.4	66.5	79.0
UI-R1-3B (Ours)	79.2	82.4	66.4	76.0
UI-R1-7B (Ours)	86.6	83.7	69.7	80.0
UI-R1-E-3B (Ours)	<u>87.0</u>	83.3	<u>71.4</u>	<u>80.6</u>
UI-R1-E-7B (Ours)	**89.2**	83.8	70.8	**81.3**

Table 3: Type (action type accuracy), GR (grounding accuracy), and SR (success rate) on ANDROIDCONTROL-LOW. Boldface shows the best; underline the second best.

Metric. Action prediction accuracy is evaluated in three dimensions:

- The **action type accuracy** evaluates the match rate between the predicted action types (e.g., `click`, `scroll`) and ground truth types;
- The **grounding accuracy** focuses specifically on the accuracy of `click` action argument predictions. We measure performance by calculating the distance between the predicted and ground truth coordinates. A prediction is considered correct if it falls within 14% of the screen size from the ground truth, following the evaluation method of UI-TARS (Qin et al. 2025);
- The **success rate** is to evaluate if a task is completely solved.

Analysis. As shown in Table 3, the comparison between UI-R1 models and Qwen2.5-VL models highlights the effectiveness of our RL-based training framework. Specifically, UI-R1-3B achieves improvements of **17.2%** in action type prediction accuracy, **8.3%** in click element grounding accuracy, and **7.1%** in success rate, using only 136 training samples. Moreover, our efficient variants, UI-R1-E-3B and UI-R1-E-7B, surpass other RL models such as GUI-R1-3B and GUI-R1-7B on low-level tasks, while requiring fewer reasoning tokens. Compared to Qwen2.5-VL models fine-tuned with supervised learning, our models demonstrate modest gains in action type accuracy and success rate, though a slight decrease in grounding accuracy is observed, which may be attributable to inconsistencies in grounding evaluation methodologies.

Key Factor Study

Reasoning Length. As shown in Table 4, UI-R1-3B not only improves accuracy over Qwen2.5-VL-3B, but also significantly reduces response length, especially in failed tasks. This reduction helps prevent the generation of irrelevant or unnecessary content, thereby enhancing reasoning efficiency. Furthermore, UI-R1-E-3B achieves even shorter responses while further increasing prediction accuracy, demonstrating that excessive reasoning length is not essential for effective grounding.

Level	Model	Response Length↓ success	failure	avg	Acc↑
Easy	Qwen2.5-VL-3B	493	1270	765	65.0
	UI-R1-3B	465 *(-28)*	491 *(-779)*	469 *(-296)*	83.3 *(+18.3)*
	UI-R1-E-3B	**113** *(-380)*	**122** *(-1148)*	**114** *(-651)*	**89.2** *(+24.2)*
Hard	Qwen2.5-VL-3B	611	775	755	11.8
	UI-R1-3B	605 *(-6)*	682 *(-93)*	661 *(-94)*	17.8 *(+6.0)*
	UI-R1-E-3B	**114** *(-497)*	**128** *(-647)*	**123** *(-652)*	**33.5** *(+21.7)*

Table 4: Response length and grounding accuracy comparison on ScreenSpot (Easy) and ScreenSpot-Pro (Hard).

As illustrated in Figure 2(a), accuracy declines as the reasoning length increases, reflecting the rising complexity of questions (as indicated by the blue histogram). Notably, the integration of reinforcement learning in UI-R1 yields substantial accuracy improvements, particularly on more complex questions (as shown by the pink polyline), highlighting the model's enhanced reasoning capabilities.

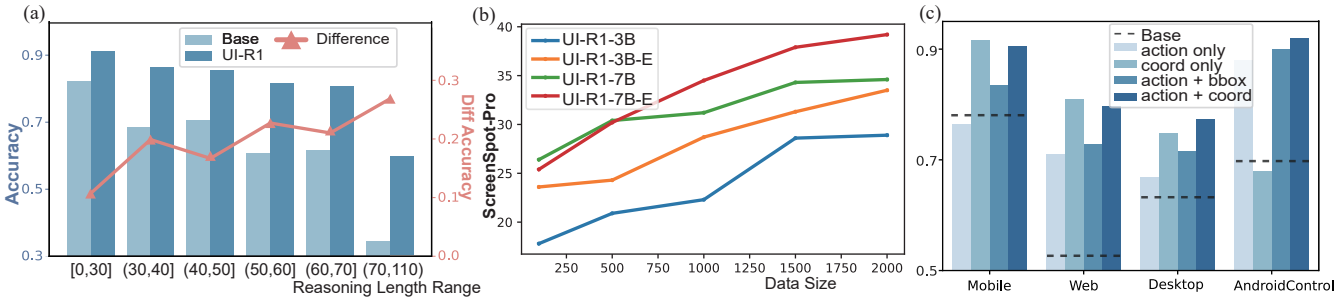

Figure 2: Key factor and ablation studies. (a) Study of relation between answering accuracy and reasoning length; (b) Ablation of training data size; (c) Ablation on reward function; All studies are evaluated on all 3 subsets of ScreenSpot.

Data Size. In Figure 2(b), we investigate the impact of training data size on model performance. As the amount of training data increases, performance on ScreenSpot-Pro improves before plateauing, indicating the model's scalability to larger datasets. Moreover, given the same model size, UI-R1-E consistently outperforms the UI-R1 and demonstrates superior scalability.

Ablation Study

Reward Function. The design of the reward function plays a crucial role in enabling the self-learning capabilities of the model. To evaluate this, we first examine the necessity of the two components of the reward function, `action + coord`. Specifically, the `action` reward improves action prediction accuracy, while the `coord` reward enhances the model's ability to ground `click` elements. Next, we compare this with an alternative reward design, `action + bbox`, where the coordinate reward $\boldsymbol{R_C}$ is replaced by an IoU-based [1] reward $\boldsymbol{R}_{\text{IoU}}$ in Equation 3.

Through ablation studies, as shown in Figure 2.(c), we demonstrate the superior effectiveness of $\boldsymbol{R_C}$ over $\boldsymbol{R}_{\text{IoU}}$ for improving `click` element grounding. However, we also observe that the action reward does not always positively impact grounding tasks. This is likely because a larger action space can introduce ambiguity, making it harder for the model to focus solely on element grounding tasks. These findings highlight the importance of carefully balancing the reward components according to the specific objectives of the task.

Training Paradigm. The corresponding ablation studies are shown in Figure 3. We find that adopting a two-stage training paradigm (first DAST to address varying task difficulties, then NOTHINK to train without explicit reasoning) promotes the more effective development of efficient reasoning strategies. We could draw to a conclusion that **Reasoning is not critical for simpler tasks such as GUI grounding**.

[1] IoU metric is calculated between the ground truth bounding box and the predicted box, and $\boldsymbol{R}_{\text{IoU}}$ assigns a value of 1 if IoU $>$ 0.5 and 0 otherwise.

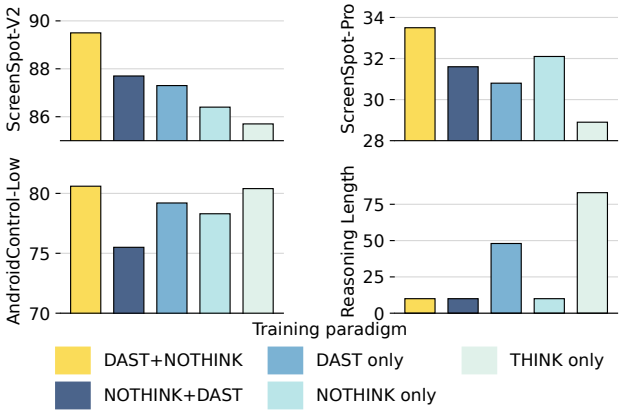

Figure 3: Ablation of thinking-or-not paradigms (DAST, NOTHINK or THINK) on ScreenSpot-V2, ScreenSpot-Pro and AndroidControl. Training epochs are all set as 8 and experiments are all done on Qwen2.5-VL-3B.

Conclusion

In this work, we present UI-R1, the first framework to enhance the reasoning capabilities of multimodal large language models (MLLMs) for GUI action prediction through rule-based reinforcement learning. To further boost inference efficiency and model performance, we propose UI-R1-E, a two-stage training paradigm that significantly reduces reasoning length without compromising accuracy.

Limitations. Although UI-R1 demonstrates promising generalization across various Human-Computer Interaction scenarios, particularly in static environments, its current design is still limited in handling the full diversity and complexity of dynamic environments.

References

Aggarwal, P.; and Welleck, S. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Anthropic. 2024. Claude Computer Use. Available at: https://www.anthropic.com/news/developing-computer-use.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Chai, Y.; Huang, S.; Niu, Y.; Xiao, H.; Liu, L.; Zhang, D.; Gao, P.; Ren, S.; and Li, H. 2024. Amex: Android multi-annotation expo dataset for mobile gui agents. *arXiv preprint arXiv:2407.17490*.

Chen, L.; Li, L.; Zhao, H.; Song, Y.; and Vinci. 2025. R1-V: Reinforcing Super Generalization Ability in Vision-Language Models with Less Than $3. https://github.com/Deep-Agent/R1-V. Accessed: 2025-02-02.

Chen, Z.; Luo, X.; and Li, D. 2025. VisRL: Intention-Driven Visual Perception via Reinforced Reasoning. *arXiv preprint arXiv:2503.07523*.

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

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

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14281–14290.

Huang, W.; Jia, B.; Zhai, Z.; Cao, S.; Ye, Z.; Zhao, F.; Hu, Y.; and Lin, S. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Li, K.; Meng, Z.; Lin, H.; Luo, Z.; Tian, Y.; Ma, J.; Huang, Z.; and Chua, T.-S. 2025a. ScreenSpot-Pro: GUI Grounding for Professional High-Resolution Computer Use.

Li, M.; Zhong, J.; Zhao, S.; Lai, Y.; and Zhang, K. 2025b. Think or Not Think: A Study of Explicit Thinking in Rule-Based Visual Reinforcement Fine-Tuning. *arXiv preprint arXiv:2503.16188*.

Li, W.; Bishop, W.; Li, A.; Rawles, C.; Campbell-Ajala, F.; Tyamagundlu, D.; and Riva, O. 2024a. On the Effects of Data Scale on Computer Control Agents. *arXiv preprint arXiv:2406.03679*.

Li, Y.; Zhang, C.; Yang, W.; Fu, B.; Cheng, P.; Chen, X.; Chen, L.; and Wei, Y. 2024b. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.

Liu, W.; Liu, L.; Guo, Y.; Xiao, H.; Lin, W.; Chai, Y.; Ren, S.; Liang, X.; Li, L.; Wang, W.; et al. 2025a. Llm-powered gui agents in phone automation: Surveying progress and prospects.

Liu, Z.; Sun, Z.; Zang, Y.; Dong, X.; Cao, Y.; Duan, H.; Lin, D.; and Wang, J. 2025b. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*.

Lu, Y.; Yang, J.; Shen, Y.; and Awadallah, A. 2024. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*.

Luo, R.; Wang, L.; He, W.; and Xia, X. 2025. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*.

Meng, F.; Du, L.; Liu, Z.; Zhou, Z.; Lu, Q.; Fu, D.; Shi, B.; Wang, W.; He, J.; Zhang, K.; et al. 2025. MM-Eureka: Exploring Visual Aha Moment with Rule-based Large-scale Reinforcement Learning. *arXiv preprint arXiv:2503.07365*.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. arXiv:2501.19393.

OpenAI. 2024. Introducing GPT-4o. Available at: https://openai.com/index/hello-gpt-4o.

Penedo, G.; Kydlíček, H.; Lozhkov, A.; Mitchell, M.; Raffel, C. A.; Von Werra, L.; Wolf, T.; et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37: 30811–30849.

Peng, Y.; Zhang, G.; Zhang, M.; You, Z.; Liu, J.; Zhu, Q.; Yang, K.; Xu, X.; Geng, X.; and Yang, X. 2025. LMM-R1: Empowering 3B LMMs with Strong Reasoning Abilities Through Two-Stage Rule-Based RL. *arXiv preprint arXiv:2503.07536*.

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

Rawles, C.; Li, A.; Rodriguez, D.; Riva, O.; and Lillicrap, T. 2023. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36: 59708–59728.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shen, H.; Zhang, Z.; Zhao, K.; Zhang, Q.; Xu, R.; and Zhao, T. 2025a. VLM-R1: A stable and generalizable R1-style Large Vision-Language Model. https://github.com/om-ai-lab/VLM-R1. Accessed: 2025-02-15.

Shen, Y.; Zhang, J.; Huang, J.; Shi, S.; Zhang, W.; Yan, J.; Wang, N.; Wang, K.; and Lian, S. 2025b. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*.

Tang, F.; Xu, H.; Zhang, H.; Chen, S.; Wu, X.; Shen, Y.; Zhang, W.; Hou, G.; Tan, Z.; Yan, Y.; Song, K.; Shao, J.; Lu, W.; Xiao, J.; and Zhuang, Y. 2025. A Survey on (M)LLM-Based GUI Agents. arXiv:2504.13865.

Wang, J.; Xu, H.; Jia, H.; Zhang, X.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024a. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*.

Wang, J.; Xu, H.; Ye, J.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024b. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*.

Wang, M.; Li, Y.; Wang, H.; Zhang, X.; Xu, N.; Wu, B.; Huang, F.; Yu, H.; and Mao, W. 2025a. Think on your Feet: Adaptive Thinking via Reinforcement Learning for Social Agents. *arXiv preprint arXiv:2505.02156*.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024c. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Wang, W.; Gao, Z.; Chen, L.; Chen, Z.; Zhu, J.; Zhao, X.; Liu, Y.; Cao, Y.; Ye, S.; Zhu, X.; et al. 2025b. VisualPRM: An Effective Process Reward Model for Multimodal Reasoning. *arXiv preprint arXiv:2503.10291*.

Wu, Z.; Cheng, P.; Wu, Z.; Ju, T.; Zhang, Z.; and Liu, G. 2025. Smoothing Grounding and Reasoning for MLLM-Powered GUI Agents with Query-Oriented Pivot Tasks. arXiv:2503.00401.

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

Xiao, W.; Gan, L.; Dai, W.; He, W.; Huang, Z.; Li, H.; Shu, F.; Yu, Z.; Zhang, P.; Jiang, H.; et al. 2025. Fast-Slow Thinking for Large Vision-Language Model Reasoning. *arXiv preprint arXiv:2504.18458*.

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

Yang, Y.; Wang, Y.; Li, D.; Luo, Z.; Chen, B.; Huang, C.; and Li, J. 2024. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*.

Zeng, W.; Huang, Y.; Liu, W.; He, K.; Liu, Q.; Ma, Z.; and He, J. 2025. 7B Model and 8K Examples: Emerging Reasoning with Reinforcement Learning is Both Effective and Efficient. https://hkust-nlp.notion.site/simplerl-reason. Notion Blog.

Zhang, C.; Yang, Z.; Liu, J.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2023. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*.

Zhou, H.; Li, X.; Wang, R.; Cheng, M.; Zhou, T.; and Hsieh, C.-J. 2025. R1-Zero's" Aha Moment" in Visual Reasoning on a 2B Non-SFT Model. *arXiv preprint arXiv:2503.05132*.