History-Aware Reasoning for GUI Agents

Ziwei Wang[1,2], Leyang Yang[1,2], Xiaoxuan Tang[3], Sheng Zhou [1*], Dajun Chen [3], Wei Jiang [3], Yong Li [3*]

[1]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, Zhejiang University
[2]College of Computer Science and Technology, Zhejiang University
[3]Ant Group
{wangziwei98, yangleyang, zhousheng_zju}@zju.edu.cn
{tangxiaoxuan.txx, chendajun.cdj, liyong.liy, jonny.jw}@antgroup.com

Abstract

Advances in Multimodal Large Language Models have significantly enhanced Graphical User Interface (GUI) automation. Equipping GUI agents with reliable episodic reasoning capabilities is essential for bridging the gap between users' concise task descriptions and the complexities of real-world execution. Current methods integrate Reinforcement Learning (RL) with System-2 Chain-of-Thought, yielding notable gains in reasoning enhancement. For long-horizon GUI tasks, historical interactions connect each screen to the goal-oriented episode chain, and effectively leveraging these clues is crucial for the current decision. However, existing native GUI agents exhibit weak short-term memory in their explicit reasoning, interpreting the chained interactions as discrete screen understanding, i.e., unawareness of the historical interactions within the episode. This history-agnostic reasoning challenges their performance in GUI automation. To alleviate this weakness, we propose a History-Aware Reasoning (HAR) framework, which encourages an agent to reflect on its own errors and acquire episodic reasoning knowledge from them via tailored strategies that enhance short-term memory in long-horizon interaction. The framework mainly comprises constructing a reflective learning scenario, synthesizing tailored correction guidelines, and designing a hybrid RL reward function. Using the HAR framework, we develop a native end-to-end model, HAR-GUI-3B, which alters the inherent reasoning mode from history-agnostic to history-aware, equipping the GUI agent with stable short-term memory and reliable perception of screen details. Comprehensive evaluations across a range of GUI-related benchmarks demonstrate the effectiveness and generalization of our method.

Code — https://github.com/BigTaige/HAR-GUI
Extended version — https://arxiv.org/abs/2511.09127

1 Introduction

Graphical User Interface (GUI) agents have witnessed remarkable advancements with the integration of advanced Multimodal Large Language Models (MLLMs), enabling autonomous manipulation of end-user devices via tailored development for GUI scenarios (Qin et al. 2025). Such capability holds significant value in applications like accessibility and automated testing. Early methods rely on MLLMs with generic multimodal understanding capability, using function calls and context engineering to manually construct workflows for GUI automation (Wang et al. 2024; Chen et al. 2024a). However, these methods demanded meticulous prompting design and faced performance bottlenecks due to models' sensitivity to instructions (Zhuo et al. 2024). Reliance on expert experience and handcrafted instructions limits scalability across GUI-oriented tasks in device ecosystems, and the closed-source nature of these large-scale MLLMs further restricts domain-specific optimization, underscoring the need for more adaptable solutions.

In contrast to screen understanding tasks (Li et al. 2020; Hsiao et al. 2022; Schoop et al. 2022; Chen et al. 2021), GUI automation is far more challenging, a GUI agent must integrate reliable reasoning with the ability to interpret a concise task description and then interact with the device step-by-step (Tang et al. 2025). In long-horizon GUI tasks, historical interaction information connects each screen to the entire episode chain shaped by the user's overall goal, and the agent's ability to reliably perceive this information is crucial for action decision at the current screen status. By integrating GUI-specific knowledge, current MLLMs (Bai et al. 2025; Team et al. 2025) can adapt to GUI scenarios, and post-training these domain-specific foundation models for GUI agent development has produced promising results (Wang et al. 2025), particularly when adopting a System-2 reasoning mode (Qin et al. 2025), in which GUI agents perform explicit logical reasoning before predicting actions, and applying RL (Shao et al. 2024) to further enhance their reasoning capability. However, despite the remarkable gains in screen perception (Zhou et al. 2025) and reasoning enhancement achieved by existing methods (Luo et al. 2025), we find that current native GUI agents exhibit a notable short-term memory weakness during episodic reasoning. Specifically, **their reasoning mode is agnostic to historical interactions, degrading chained interactions to discrete screen understanding**. This "history-agnostic" stems from the foundation MLLM's inherent Chain-of-Thought (CoT), undermining performance in episodic reasoning that need to leverage the previous execution clues.

Current advanced methods focus on enhancing the overall reasoning of GUI agents, whereas exploration of spe-

*Corresponding Author

cific defects in episodic reasoning remains limited. Drawing on heuristic learning, several methods leverage hard samples, i.e., instances that are previously incorrectly inferred, and apply RL for reflective training, encouraging the agent to acquire unfamiliar domain-specific reasoning knowledge from these samples (Qin et al. 2025; Liu et al. 2025b). RL primarily steers the model toward reward-bearing trajectories, thereby refining its output strategy from generating multiple candidates (pass@k) to a single, high-confidence solution (pass@1). However, this process remains fundamentally limited by the foundation model's prior knowledge. (Yue et al. 2025). Without introducing external GUI-specific reasoning knowledge, the agent's short-term memory weakness persists. Moreover, these methods perform RL with inference-format instructions (Lu et al. 2025; Wu et al. 2025a,b), i.e., training instructions identical to the inference phase, yet our experiments indicate that the optimization signal primarily affects action-level prediction without enhancing the agent's inherent reasoning mode.

In this work, we aim to enhance the reasoning capabilities of the GUI agent by equipping it with short-term memory for episodic reasoning, enabling explicit historical-context awareness in its System-2 CoT, which in turn fortifies its overall performance on GUI-oriented tasks. Consequently, we propose a **H**istory-**A**ware **R**easoning **(HAR)** framework for reflective training with tailored GUI reasoning enhancement strategies. Our method comprises three key components: ***(i)* Reflective learning scenario construction**; ***(ii)* Tailored correction guidelines synthesis**; ***(iii)* A hybrid RL reward function to encourage historical awareness in the GUI agent**. Using the HAR framework, we alter the reasoning mode of the GUI agent from being history-agnostic to being history-aware. This allows for explicit integration of historical context and cognitive correction via error-centric self-evolution. As a result, the GUI agent emerges stable short-term memory, allowing it to flexibly perceive the episode's chained historical clues and make reasonable use of it. This strengthened reasoning enables the GUI agent to handle long-horizon interactions and achieving consistent and persistent gains across GUI-oriented tasks.

We evaluate our method on diverse, widely used GUI-related benchmarks and manually annotate a challenging Chinese mini-program benchmark for out-of-distribution (OOD) generalization comparison. Experiments demonstrate that our method outperforms current advanced methods with similar parameters on multiple GUI-related benchmarks and rivals larger models in OOD scenarios. Our main contributions are as follows:

- We propose HAR, a framework that employs tailored strategies and reflective learning to deepen the agent's GUI-specific knowledge and transform its reasoning from history-agnostic to history-aware, enhancing its short-term memory for episodic reasoning.
- Using the HAR framework, we develop **HAR-GUI-3B**, a GUI-tailored native model with reliable episodic reasoning and screen perception, and demonstrate its effectiveness across a range of GUI-related benchmarks.
- HAR-GUI-3B shows consistent generalization. In OOD scenarios, it outperforms SOTA methods with similar parameter sizes and competes with much larger models.

2 Related Works

Early methods rely on sophisticated context engineering and MLLM function calls to construct workflows for GUI automation, *e.g.*, ReAct (Yao et al. 2023), Reflexion (Shinn et al. 2023), Expel (Zhao et al. 2024) and AppAgent (Zhang et al. 2025). While these methods show promise, their effectiveness relies on the MLLM's general capability, manual experience, and sensitivity to instructions. Subsequently, researchers enhance MLLMs' GUI perception by training small-parameter models with domain-specific data, yielding native GUI agents such as SeeClick (Cheng et al. 2024), MP-GUI (Wang et al. 2025) and ShowUI (Lin et al. 2025) via supervised fine-tuning (SFT). With RL's success in reasoning enhancement (Shao et al. 2024), several methods employ RL to train agents (Bai et al. 2024), achieving remarkable reasoning and generalization capabilities. Domain-specific pre-training further enables MLLMs to tackle complex GUI tasks using concise instructions (Bai et al. 2025). The slow thinking mode of System-2 reasoning emerges as an effective approach for enhancing GUI agents (*e.g.*, UI-TARS (Qin et al. 2025), InfiGUI-R1-3B (Liu et al. 2025b), GUI-R1-3B (Luo et al. 2025) and UI-R1-3B (Lu et al. 2025)). Meanwhile, several methods steer the agent's attention to specific content via instructional constraints (Wang et al. 2024; Zhang et al. 2025). However, this manual prompting approach contradicts the natural reasoning behavior of the foundation models on GUI automation, resulting in unstable performance and hallucinations. Alternatively, we construct a reflective learning scenario and tailored strategies to equip the agent with competent GUI reasoning capability.

3 HAR Framework

An overview of the Histoty-Aware Reasoning (HAR) framework is illustrated in Fig.1. In this section, we first define the GUI episode reasoning task and then introduce the critical training stages in the HAR framework.

3.1 Task Definition

We first formulate the execution of the GUI agent, let $\mathcal{D}$ denotes an episode chain with an overall goal $\mathcal{G}$, $\mathcal{D} = (\mathcal{G}, (O_1, \mathcal{A}_1), \ldots, (O_n, \mathcal{A}_n))$, and observation O_t=$(I_t, \mathbb{P})$, where $\mathcal{A}_t \in \mathbb{A}$ means the action executed by the agent at time step $t \in [1, n]$, $\mathbb{A}$ is the pre-defined action space (*e.g.*, `CLICK`, `SCROLL` and `TYPE`), I_t is the screen image, $\mathbb{P}$ denotes the textual instruction template. The task can be formulated as a Markov Decision Process $P(\mathcal{A}_t|O_{\leq t}, A_{<t}, \mathcal{G})$. Since step-by-step execution, chained interaction histories $\mathscr{T}_{<t}$ are critical for the current decision. While some methods concatenate $(I_{<t}, \mathcal{A}_{<t})$ pairs into the input for $\mathscr{T}_{<t}$ transmission, computing image tokens is expensive, especially for high-resolution devices (Lin et al. 2025). To balance computational costs and performance, most research conveys $\mathscr{T}_{<t}$ via textual modality (Liu et al. 2025b; Luo et al. 2025) by summarizing interactions and integrating them into $\mathbb{P}$, which is also a strategy adopted in this work.

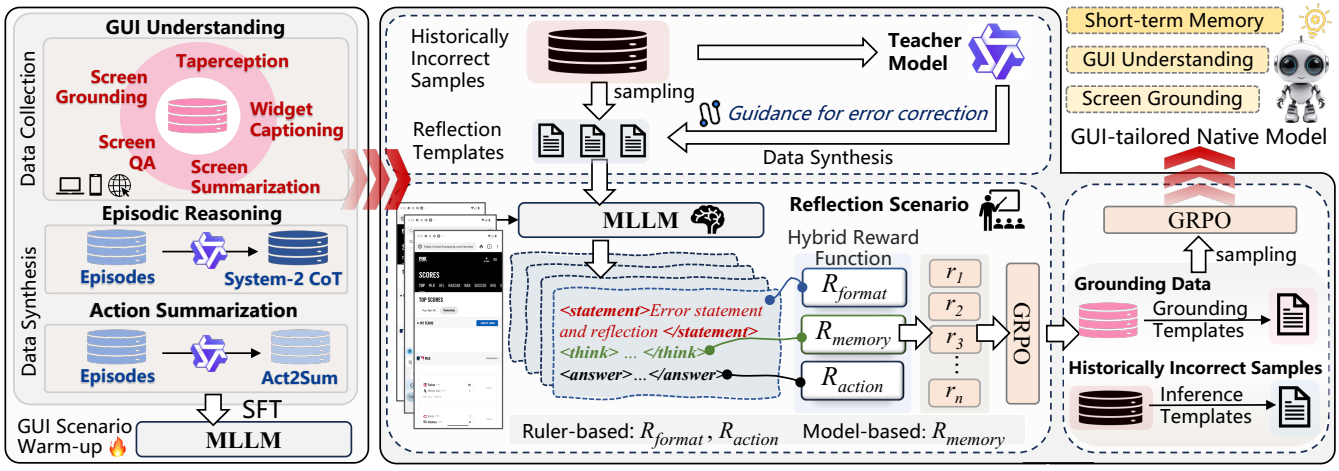

Figure 1: Overview of the Histoty-Aware Reasoning (HAR) framework. HAR framework is an error-centric learning approach designed to enhance the reasoning capability of the GUI agent by performing error-aware cognitive correction within a tailored reflection scenario. The framework consists of two critical training stages: ***(i)* GUI Scenario Warm-up Stage.** During this phase, comprehensive domain-specific knowledge is injected into the agent via GUI-related data collection and synthesis for knowledge distillation. ***(ii)* Learning From Failure Stage.** In this stage, the agent's short-term memory is enhanced. It involves a round of RL within the reflection scenario to perform error-aware cognitive corrections that boost episodic reasoning, followed by another round of RL employing a task-mixing training strategy to assist the GUI agent perceive screen visual details.

3.2 GUI Scenario Warm-up

Due to the weak screen perception of the foundation model $\mathcal{M}_{base}$ in handling GUI-oriented tasks, in this training stage, we collect and synthesize comprehensive GUI-related data for the injection of domain-specific knowledge through supervised fine-tuning (SFT).

GUI Understanding Enhancement. Effective screen perception is crucial for GUI agents to conduct GUI-oriented tasks (Cheng et al. 2024; Wang et al. 2025), *e.g.*, screen analysis (Li et al. 2020; Wang et al. 2021) and screen question-answering (Baechler et al. 2024). To enhance the fundamental screen understanding of $\mathcal{M}_{base}$, we collect comprehensive domain-specific tasks from public sources, including graphic caption generation, UI-element clickability prediction, screen question answering, screen summary, and screen grounding. These data cover platforms with multiple resolutions, including mobile, desktop, and website.

Action Summarization Integration. The GUI agent is required to interact with the screen for multiple rounds, analyze the current screen status, and interpret the overall goal to make the ongoing decision. Thus, the agent needs to be equipped with a reliable action understanding capability, i.e., it is crucial for the agent to effectively comprehend the semantics of the action at the episode level.

In this part, we introduce an action summarization task designed to enhance the action semantic understanding of $\mathcal{M}_{base}$. Specifically, we construct instruction templates that prompt a teacher model to synthesize **Action-to-Summary (Act2Sum)** data. Specifically, let $\mathcal{G}$ denote the goal, $\mathcal{I}_t$ the current screen image, $\mathcal{A}_t$ the corresponding action, $\mathbb{P}_{sum}$ the instruction template, and $\mathcal{M}_{teacher}(\cdot, \theta)$ the teacher model with parameters θ. The process can be expressed as follows,

$$O^t_{Act2Sum} = \mathcal{M}_{teacher}(\mathcal{I}_t, \mathbb{P}_{sum}(\mathcal{G}, \mathcal{A}_t), \theta) \quad (1)$$

where $O^t_{Act2Sum}$ is the action summary corresponding to $\mathcal{A}_t$, $\mathcal{I}_t$, and $\mathcal{G}$. We construct the Act2Sum dataset by pairing each summary $O^t_{Act2Sum}$ with its input $(\mathcal{I}_t, \mathbb{P}_{sum}(\mathcal{G}, \mathcal{A}_t))$.

Since the input explicitly contains the episode's goal, the synthesized content is goal-oriented and semantically rich, which assists $\mathcal{M}_{base}$ in generating globally aware historical interaction records during episodic reasoning.

System-2 Reasoning Data Distillation. Compared with System-1 reasoning paradigm (*fast thinking mode that generates answers directly*), explicitly generate intermediate thought processes via CoT before answering, System-2 reasoning (*slow thinking*) denotes intentional, organized, and analytical thought, improving agents to tackle complex and multi-step tasks (Qin et al. 2025). To enhance the reasoning capability of $\mathcal{M}_{base}$, we utilize the reasoning advanced $\mathcal{M}_{teacher}$ to synthesize System-2 CoT for each instance of episodes. Specifically, we build the instruction template (*inference format*) and conduct the inference with $\mathcal{M}_{teacher}$. Afterwards, we filter out positive samples through evaluation methods and collect these synthetic samples with System-2 CoT as training data. Via training, we achieve high-quality GUI reasoning knowledge injection from $\mathcal{M}_{teacher}$ to $\mathcal{M}_{base}$.

3.3 Learning From Failure

After the warm-up stage, we yield a GUI-enhanced agent $\mathcal{M}_{warm\text{-}up}$ from $\mathcal{M}_{base}$. Inspired by the heuristic principle that *"Adversity is the crucible of growth"*, we construct a reflection scenario for training to encourage $\mathcal{M}_{warm\text{-}up}$ to grasp domain-sepcific reasoning knowledge from hard samples via self-evolution, thereby improving the reasoning of

GUI agent when facing complex GUI-oriented tasks. We first perform inference using $\mathcal{M}_{warm\text{-}up}$ on our episode data and flag error instances as historical incorrect samples $\mathbb{D}_{his}$.

Guidance Synthesis for Error Correction. Currently, there are few methods conduct RL for reflective training on hard samples (Wu et al. 2025a; Liu et al. 2025a). However, these methods are trained under an inference-style pattern. Although RL can narrow the output space from pass@k to pass@1 to improve the success rate through exploration (Yue et al. 2025), enhancing short-term memory remains a challenge. This requires altering the GUI agent's entrenched reasoning CoT established during pre-training.

Inspired by curriculum learning (Liu et al. 2024), we conduct error analysis on samples in $\mathbb{D}_{his}$ by prompting $\mathcal{M}_{teacher}$ to generate no more than three error correction guidelines $\mathbb{G}$ for each incorrect instance. Thus, $\mathbb{G}$ can serve as external GUI reasoning knowledge to facilitate $\mathcal{M}_{warm\text{-}up}$ in clue-oriented reasoning enhancement and hard samples correction. We can formulate this process as follows,

$$\begin{aligned} &\mathbb{G}_t = \mathcal{M}_{teacher}(\mathcal{I}_t, \mathbb{P}_{guidance}(\hat{x}_t), \theta), \\ &\hat{x}_t = (\mathcal{G}, \mathscr{T}_{<t}, \mathcal{A}_t, \mathcal{A}_t^{error}, \mathcal{C}_t^{error}) \in \mathbb{D}_{his} \end{aligned} \tag{2}$$

where $\mathbb{G}_t$ means the tailored guidelines corresponding to x_t, $\mathcal{A}_t$ is the ground truth, $\mathbb{P}_{guidance}(\cdot)$ represents the instruction, $\mathcal{A}_t^{error}$ denotes the incorrect action conducted by $\mathcal{M}_{warm\text{-}up}$ and $\mathcal{C}_t^{error}$ is the CoT corresponding to $\mathcal{A}_t^{error}$.

Reflection Scenario Construction. Inspired by earlier methods that employ MLLM function calls and handcrafted, prompt-based reflection to correct erroneous execution paths (Yao et al. 2023; Wang et al. 2024), current approaches adopt inference-format instructions and perform RL training on hard samples to enhance GUI agents (Wu et al. 2025a; Liu et al. 2025a). As discussed, the weak domain-specific knowledge of foundation models in GUI automation results in a key observation: RL-based exploration under the inference-format instruction struggles both to correct hard samples and to enhance the short-term memory of GUI agents. Thus, we construct a reflection scenario for cognitive correction and introduce external GUI reasoning knowledge tailored for each instance in $\mathbb{D}_{his}$, i.e., guidelines $\mathbb{G}$, to assist the GUI agent in its self-evolution during clue-guided exploration. The process can be formulated as,

$$\begin{aligned} &\mathcal{O}_t^* = \mathcal{M}_{warm\text{-}up}(\mathcal{I}_t, \mathbb{P}_{reflection}(x_t), \theta), \\ &x_t = (\mathcal{G}, \mathbb{G}_t, \mathscr{T}_{<t}, \mathcal{A}_t^{error}, \mathcal{C}_t^{error}) \in \mathbb{R}_{his} \end{aligned} \tag{3}$$

where $\mathcal{A}_t^*, \mathcal{C}_t^*, \mathcal{S}_t^* = \mathcal{O}_t^*$, $\mathbb{P}_{reflection}$ denotes the instruction of the reflection format, $\mathcal{A}_t^*$ represents the action predicted by $\mathcal{M}_{warm\text{-}up}$, $\mathcal{C}_t^*$ and $\mathcal{S}_t^*$ mean the corresponding CoT, as well as the statement and reflection on the previous incorrect prediction, respectively.

Error-Aware Cognitive Correction. Since we aim to guide the GUI agent to grasp episodic reasoning knowledge through self-exploration without supervisory signal constraints, we identify that the GRPO algorithm (Shao et al. 2024) aligns well with this goal. Specifically, GRPO first generates N candidate responses $\{\mathcal{O}_{t,i}^*\}_{i=1:N}$ for a task query. Each response $\mathcal{O}_{t,i}^*$ yields a reward r_i via action execution. Then, normalizing rewards to compute the relative advantage E_i of each response:

$$E_i = \frac{r_i - mean(\{r_1, \ldots, r_N\})}{std(\{r_1, \ldots, r_N\})} \tag{4}$$

where $mean(\cdot)$ and $std(\cdot)$ denote the reward distribution's mean and standard deviation. In our reflection scenario, each response $\mathcal{O}_{t,i}^*$ has three components: $\mathcal{A}_{t,i}^*$, $\mathcal{C}_{t,i}^*$, and $\mathcal{S}_{t,i}^*$. To make fuller utilize of the output signals, each sample's reward r_i consists of multiple parts. First, a rule-based format reward r_i^{format} is used to check whether $\mathcal{O}_{t,i}^*$ conforms to the output format required by instruction $\mathbb{P}_{reflection}$. If it matches, $r_i^{format} = 1$, otherwise it is 0. Next, we define the action reward r_i^{action}. If the predicted action $\mathcal{A}_{t,i}^*$ matches the label, the reward is 1, otherwise it is 0.

Note that in real-world execution, actions involving screen coordinates are frequently called, and our experiments found that the correct execution of such actions (*e.g.*, `CLICK`) is challenging for GUI agents. Thus, for this type of actions, we assign a higher reward to enhance the GUI agent's perception of screen details. Specifically, let $P = (p_x, p_y)$ and $P^* = (p_x^*, p_y^*)$ represent the absolute coordinates (x, y) of the label and prediction, respectively. Then, we calculate the scalar action reward r_i^{action} as follows,

$$\begin{aligned} &\mathcal{F}_{dist}(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \\ &P_1 = (x_1, y_1), P_2 = (x_2, y_2) \end{aligned} \tag{5}$$

$$\mathcal{F}_{abs}(P_1, P_2, \tau) = \begin{cases} 1 - \mathcal{F}_{dist}(P_1, P_2)/\tau, \text{ if } \mathcal{F}_{dist} < \tau \\ 0, \text{else} \end{cases} \tag{6}$$

$$\begin{aligned} &P_{norm} = (p_x/w, p_y/h), P_{norm}^* = (p_x^*/w, p_y^*/h) \\ &\mathcal{D}_{norm} = \mathcal{F}_{dist}(P_{norm}, P_{norm}^*) \\ &r_i^{action} = \begin{cases} 1 + \mathcal{F}_{abs}(P, P^*, \tau_{abs}^1), \text{ if } \mathcal{D}_{norm} \le \tau_{norm} \\ \mathcal{F}_{abs}(P, P^*, \tau_{abs}^2), \text{else} \end{cases} \end{aligned} \tag{7}$$

where w and h denote screen resolution, while τ_{norm}, τ_{abs}^1 and τ_{abs}^2 represent thresholds. This approach offers two key benefits: ***(i)* it motivates the GUI agent to optimize along the path of minimal Euclidean distance when all predictions within a group align with the label**; ***(ii)* it rewards deviations from the nearest predictions when mismatches occur**. This multi-scale reward mechanism encourages the GUI agent to explore fine-grained screen details.

Via analysis of bad cases and guidelines $\mathbb{G}$, we notice that many incorrect predictions result from the GUI agent's lack of awareness of previous interaction histories, indicating weak short-term memory. Thus, we propose a model-based **Memory-Augmented Reward (MAR)** function. Our goal is to determine whether the explicit CoT $\mathcal{C}_{t,i}^*$ includes the agent's logical analysis of previous interactions. Let r_i^{memory} denotes the MAR, which can be calculated as,

$$r_i^{memory} = \mathcal{F}_{\text{MAR}}(\mathbb{P}_{Memory}(\mathcal{C}_{t,i}^*, \mathscr{T}_{<t}^i)) \tag{8}$$

where $\mathbb{P}_{Memory}$ represents the instruction for short-term memory verification and $\mathcal{F}_{\text{MAR}}(\cdot)$ denotes a model-based reward function[1]. If CoT $\mathcal{C}_{t,i}^*$ contains historical information

[1]We use Qwen3-235B-A22B (Yang et al. 2025) as the memory reward judgment function.

$\mathcal{T}^i_{<t}$, then r_i^{memory}=1, otherwise it is 0. Note that not all guidelines $\mathbb{G}$ contain clues to the context of historical interactions. Thus, compared with statically constraining the GUI agent to focus on historical interactions within the episode, our "guidelines + MAR" strategy can dynamically guide the agent to perform error-aware cognitive correction and enhance its short-term memory. Finally, we define the hybrid reward r_i as follows, where γ is a hyperparameter,

$$r_i = r_i^{format} \times (r_i^{action} + \gamma \times r_i^{memory}). \quad (9)$$

After the first round of RL (Round-1 RL) in our constructed reflection scenario with tailored reward functions, we enhanced the short-term memory of the GUI agent in episodic reasoning, altering it from a history-agnostic and simplistic reasoning mode to a history-aware and rigorous reasoning mode. The effect of this stage is shown in Fig.2.

Round-2 RL. Since the external episodic reasoning knowledge, i.e., tailored guidelines $\mathbb{G}$, is unavailable during the execution phase, we introduce Round-2 RL to align the execution format. Through Round-1 RL in the reflection scenario, the GUI agent enhances its capability to correct erroneous cognition, including improvements in short-term memory, action semantic understanding, and screen details perception. At this stage, instructions are converted to the inference-format to raise the difficulty of learning[2]. Further, **we noticed that reflective training with episodic reasoning data alone weakens the GUI agent's grounding capability** (*as discussed in the ablation experiment,* Fig.3). To address this issue, we propose a **task mixing training strategy (TMTS)**, which is a multi-task RL approach that mixes grounding and GUI episodic reasoning tasks. For episodic reasoning task, we use the Round-1 RL hybrid reward function in Eq.9; for grounding task, we use Eq.7. See the appendix for details of each instruction template.

After completing Round-2 RL, we upgrade $\mathcal{M}_{warm\text{-}up}$ to a reasoner $\mathcal{M}_{HAR\text{-}GUI}$ (HAR-GUI-3B) with comprehensive GUI knowledge, which can serve as a GUI-tailored native model. Experiments demonstrate the advancement of $\mathcal{M}_{HAR\text{-}GUI}$ across a range of GUI-related benchmarks.

4 Experiments

4.1 Experimental Setup

Data Curation. Overall, there are three training stages in the HAR framework. ***(i)*** We first distill GUI knowledge via the **warm-up SFT stage**, sampling 4k GUI understanding instances from MP-GUI, 20k grounding instances from OS-Atlas, and synthesizing 58k instances equipped with System-2 CoT alongside 100k Act2Sum entries from AITW, Mind2Web, and GUI-Odyssey. ***(ii)*** The **Round-1 RL stage** then conducts in the reflection scenario to perform error-aware cognitive correction. We synthesize 15k tailored guidance from hard samples to construct the reflection templates. ***(iii)*** Finally, the **Round-2 RL stage** employs our TMTS to enhance the GUI agent's screen detail perception and knowledge integration. We sample 15k grounding data from OS-Atlas and 15k instances from the pre-collected hard samples.

[2]Keeping the same instruction as when obtaining $\mathbb{D}_{his}$, and the output $\mathcal{O}^*_{t,i}$ only includes $\mathcal{A}^*_{t,i}$ and $\mathcal{C}^*_{t,i}$.

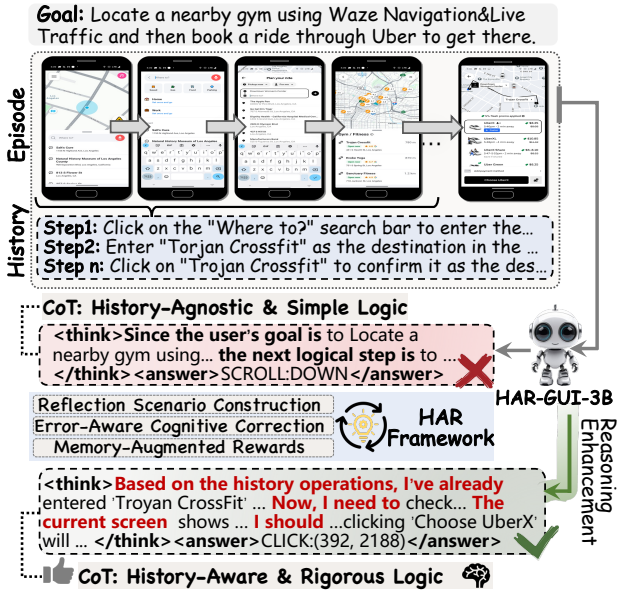

Figure 2: Short-term memory emergence and reasoning enhancement through HAR framework.

Benchmarks. We extensively evaluate the proposed HAR-GUI-3B on three types of widely used GUI-related benchmarks, ***(i)*** **GUI episodic reasoning benchmarks** include AITW (Rawles et al. 2023), Mind2Web (Deng et al. 2023), GUI-Odyssey (Lu et al. 2024a), and a manually collected in-house data for OOD evaluation, ***(ii)*** **GUI grounding benchmarks** include ScreenSpot (Cheng et al. 2024) and ScreenSpot-V2 (Li et al. 2025), and ***(iii)*** **GUI understanding benchmarks** include ScreenQA (QA) (Hsiao et al. 2022), ScreenQA Short (QAS) (Baechler et al. 2024), Complex ScreenQA (CQA) (Baechler et al. 2024), WebSRC (WS) (Chen et al. 2021), Screen2Words (S2W) (Wang et al. 2021), Taperception (TP) (Schoop et al. 2022) and Widget Captioning (WC) (Li et al. 2020).

Implementation Details. Our HAR-GUI-3B is built upon Qwen2.5-VL-3B-Instruct and is trained using the proposed HAR framework. We train the warm-up SFT stage for 1 epoch with a learning rate of 5e-6 and a global batch size of 256. For both Round-1 RL and Round-2 RL stages, we train for 2 epochs with a learning rate of 2e-6 and a batch size of 32. We set hyperparameters as τ_{norm}=0.1, τ^1_{abs}=40, τ^2_{abs}=200, and γ=0.2. LoRA with rank 64 and alpha 128 is applied to the LLM and Vision Backbone, using AdamW as optimizer. After training via the HAR framework, we perform post-training (SFT) of HAR-GUI-3B on GUI understanding (Wang et al. 2025) and GUI episodic reasoning benchmarks individually with a learning rate of 2e-5 and batch size of 256, train each task for 4 epochs, and record the best results. Baselines use the same training settings as HAR-GUI-3B. Qwen2.5-VL-72B-Instruct serves as the teacher model for all data synthesis. All experiments are performed on 8 NVIDIA A100 80GB GPUs.

Method	General	Install	G.Apps	Single	WebShop.	Overall
OmniParser (Lu et al. 2024b)	48.3	57.8	51.6	77.4	52.9	57.7
SeeClick (Cheng et al. 2024)	54.0	66.4	54.9	63.5	57.6	59.3
UI-R1-3B (Lu et al. 2025)	54.3	63.6	58.6	68.2	54.9	59.9
InternVL2-8B (Chen et al. 2024b)	58.1	65.3	56.8	68.7	61.1	62.0
R-VLM (Park et al. 2025)	59.9	70.6	59.6	72.5	61.7	64.9
Qwen2.5-VL-3B (Bai et al. 2025)	61.2	69.8	62.9	70.8	62.4	65.4
GUI-R1-3B (Luo et al. 2025)	59.2	68.9	64.5	71.9	63.2	65.6
InfiGUI-R1-3B (Liu et al. 2025b)	62.6	72.3	66.4	72.3	64.9	67.7
ShowUI (Lin et al. 2025)	63.5	72.3	66.0	72.3	65.8	68.3
MP-GUI (Wang et al. 2025)	63.7	**74.3**	65.3	75.4	**67.2**	69.2
HAR-GUI-3B	**63.8**	73.6	**69.5**	**77.9**	66.1	**70.2**

Table 1: Performance comparison on AITW. The evaluation metric used is step success rate (SSR).

Method	Cross-Task			Cross-Website			Cross-Domain		
	Acc.	F1	SSR	Acc.	F1	SSR	Acc.	F1	SSR
OmniParser (Lu et al. 2024b)	42.4	87.6	39.4	41.0	84.8	36.5	45.5	85.7	42.0
ShowUI (Lin et al. 2025)	39.7	88.0	36.9	41.0	83.6	34.2	38.9	85.3	34.1
UI-R1-3B (Lu et al. 2025)	42.4	85.8	36.8	44.4	83.1	36.7	43.0	83.7	36.3
Qwen2.5-VL-3B (Bai et al. 2025)	42.0	87.9	39.0	45.5	84.6	37.6	43.2	84.9	37.9
InfiGUI-R1-3B (Liu et al. 2025b)	41.9	86.5	37.2	42.7	84.9	37.7	43.6	83.1	38.2
GUI-R1-3B (Luo et al. 2025)	42.3	85.5	38.8	45.8	84.2	38.5	44.7	85.9	38.9
MP-GUI (Wang et al. 2025)	42.1	89.0	38.1	39.4	87.1	32.9	37.6	87.4	33.7
HAR-GUI-3B	**47.9**	**89.6**	**42.2**	**49.1**	**87.3**	**41.2**	**47.3**	**88.3**	**44.0**

Table 2: Performance comparison on Mind2Web. We report element accuracy (Acc.), operation F1 (F1), and SSR.

4.2 Main Results

Overall Performance. We select AITW, Mind2Web and GUI-Odyssey to evaluate the agentic performance of HAR-GUI-3B. As shown in Tab.1, Tab.2, and Tab.3, HAR-GUI-3B consistently surpasses the current advanced methods, even those with far more parameters, *e.g.* MP-GUI (8B), SeeClick (9.8B), and Qwen2.5-VL-7B. Under identical training settings, it delivers substantial gains over the foundational Qwen2.5-VL-3B. Further, when compared with GUI agents that share the same MLLM architecture, *e.g.*, UI-R1-3B, GUI-R1-3B and InfiGUI-R1-3B, HAR-GUI-3B still leads by a clear margin. **These results indicate that our HAR framework effectively enables domain-specific reasoning skills from hard samples in the constructed reflection scenario**, enabling reliable GUI automation.

Grounding. Grounding capability is crucial for GUI automation (Cheng et al. 2024), as it determines whether the GUI agent can accurately execute click-based actions on the screen. We compare the grounding performance of HAR-GUI-3B and the current advanced methods on ScreenSpot and ScreenSpot-V2. As shown in Tab.4 and Tab.5, HAR-GUI-3B delivers the leading results. We attribute this reliable screen grounding performance to the TMTS (Sect.3.3) used in Round-2 RL training stage[3]. **The mixture of the grounding task further enhances the GUI agent in screen detail perception and GUI knowledge acquisition**.

Method	Tool	Info.	Shop.	Media	Social	M.Apps	Overall
GPT-4o (Hurst et al. 2024)	20.81	16.28	31.91	15.38	21.28	16.67	20.39
Qwen2.5-VL-3B (Bai et al. 2025)	53.86	43.44	43.01	44.74	45.05	45.72	46.14
UI-R1-3B (Lu et al. 2025)	55.03	43.81	43.69	45.17	47.09	45.46	46.71
GUI-R1-3B (Luo et al. 2025)	57.46	44.87	44.71	46.67	49.42	46.97	48.35
InfiGUI-R1-3B (Liu et al. 2025b)	60.60	46.62	45.19	47.44	53.48	50.38	50.62
Qwen2.5-VL-7B (Bai et al. 2025)	70.24	57.24	49.28	58.34	60.32	54.93	58.39
HAR-GUI-3B	**74.62**	**58.53**	**51.32**	**62.19**	**65.51**	**61.70**	**62.31**

Table 3: Performance on GUI-Odyssey. The metric is SSR.

Method	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
MP-GUI (Wang et al. 2025)	86.8	65.9	70.8	56.4	58.3	46.6	64.1
UGround-7B (Gou et al. 2025)	82.8	60.3	82.5	63.6	80.4	70.4	73.3
ShowUI (Lin et al. 2025)	92.3	75.5	76.3	61.1	81.7	63.6	75.1
Qwen2.5-VL-7B (Bai et al. 2025)	93.8	72.5	87.6	65.7	88.7	70.4	79.8
UI-R1-3B (Lu et al. 2025)	–	–	90.2	59.3	85.2	73.3	–
GUI-R1-3B (Luo et al. 2025)	–	–	93.8	64.8	89.6	72.1	–
UI-TARS-2B (Qin et al. 2025)	93.0	75.5	**94.3**	68.6	84.3	**74.8**	82.3
OS-Atlas-7B (Wu et al. 2024)	93.0	72.9	91.8	62.9	**90.9**	74.3	82.5
HAR-GUI-3B	**94.5**	**81.0**	93.8	**70.8**	85.6	73.8	**83.3**

Table 4: Performance comparison on ScreenSpot.

Method	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
OS-Atlas-4B (Wu et al. 2024)	87.2	59.7	72.7	46.4	85.9	63.1	71.9
GPT-4o + OS-Atlas-4B (Wu et al. 2024)	95.5	75.8	79.4	49.3	90.2	66.5	79.1
InternVL3-8B (Zhu et al. 2025)	–	–	–	–	–	–	81.4
Qwen2.5-VL-3B (Bai et al. 2025)	95.0	80.1	90.2	64.3	88.0	70.4	81.3
OS-Atlas-7B (Wu et al. 2024)	95.2	75.8	90.7	63.6	90.6	77.3	84.1
UI-TARS-2B (Qin et al. 2025)	95.2	79.1	90.7	68.6	90.6	77.3	84.7
UI-R1-3B (Lu et al. 2025)	96.2	**84.3**	92.3	63.6	89.2	75.4	85.4
HAR-GUI-3B	**96.5**	81.0	**95.4**	**76.5**	88.8	**78.8**	**86.2**

Table 5: Performance comparison on ScreenSpot-V2.

OOD Evaluation. To evaluate the OOD generalization of current SOTA methods, we develop a challenging Chinese mobile automation benchmark (Tab.7). We manually collect and annotate 415 tasks from Alipay mini programs (`CLICK`-only action space). As reported in Tab.6, HAR-GUI-3B significantly outperforms other advanced methods with comparable parameter sizes. **This remarkable performance mainly stems from its stable short-term memory and its reliable screen perception capability.**

GUI Understanding. In this part, we compare the basic GUI understanding effect of HAR-GUI-3B. We select the comprehensive benchmarks collected by MP-GUI, which include screen analysis (WC and TP), screen question-answering (QA, CQA, QAS, and WS) and screen summarization (S2W). In Tab.8, HAR-GUI-3B outperforms advanced methods on most GUI-oriented tasks with fewer parameters. Compared to the foundational Qwen2.5-VL-3B, our HAR-GUI-3B shows an overall gain of 3.07 points with the same training settings. Compared with methods with larger parameters, HAR-GUI-3B is still competitive. In particular, our method surpasses the current SOTA GUI-specific method MP-GUI (8B) on S2W, WS, QA and QAS benchmarks, and UI-TARS-1.5-7B on S2W and TP benchmarks. **Such results demonstrate that HAR-GUI-3B can implicitly learn multi-grained screen knowledge through error-aware cognitive correction and tailored training recipe.**

[3]We find integrating the grounding task into episodic reasoning improves both tasks compared with training them sequentially.

Method	Takeout		Repast		Finance		Insurance	
	SSR	SR	SSR	SR	SSR	SR	SSR	SR
Qwen2.5-VL-72B (Bai et al. 2025)	80.60	17.65	**85.02**	**20.56**	**86.91**	**57.43**	**69.98**	**21.78**
Gemini-2.5-Pro (Gemini Team 2025)	71.96	4.76	68.83	6.74	48.81	5.62	57.83	10.23
UI-TARS-7B-DPO (Qin et al. 2025)	37.65	0.00	39.47	0.00	70.80	48.80	36.84	11.54
MiMo-VL-7B-RL (Team et al. 2025)	56.74	0.98	64.24	3.74	73.41	35.64	64.21	14.85
Qwen2.5-VL-7B (Bai et al. 2025)	46.27	0.00	69.26	3.74	59.71	20.79	56.66	16.83
GUI-R1-3B (Luo et al. 2025)	71.21	10.7	70.49	5.77	69.99	30.01	63.60	14.78
UI-R1-3B (Lu et al. 2025)	59.60	1.00	67.31	4.76	61.15	19.80	58.63	9.90
Qwen2.5-VL-3B (Bai et al. 2025)	51.21	0.00	62.01	0.00	57.08	18.19	50.51	8.24
InfiGUI-R1-3B (Liu et al. 2025b)	61.90	0.00	69.11	3.99	51.74	8.77	45.69	7.55
HAR-GUI-3B	**82.76**	**24.30**	77.69	11.76	76.50	35.60	69.19	**21.78**

Table 6: Zero-Shot comparison in OOD scenarios. We report SSR and episode-wise success rate (SR).

Takeout			Repast		
#Tasks	#Steps	Avg. Steps	#Tasks	#Steps	Avg. Steps
103	1,031	10.11	108	1,015	9.49
Finance			**Insurance**		
#Tasks	#Steps	Avg. Steps	#Tasks	#Steps	Avg. Steps
102	489	4.82	102	503	4.98

Table 7: In-house GUI episode data statistics cover 4 categories of widely used Chinese mobile app scenarios.

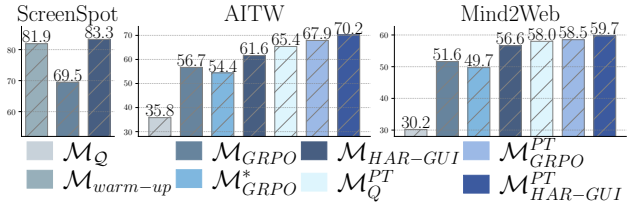

Figure 3: Effectiveness of HAR framework. $\mathcal{M}_{\mathcal{Q}}$ is Qwen2.5-VL-3B-Instruct zero-shot results. $\mathcal{M}_{GRPO}$ refers to the method keeping the same settings as $\mathcal{M}_{HAR\text{-}GUI}$, but excluding the reflection scenario (using the inference-format instruction), MAR, and TMTS. $\mathcal{M}^{PT}_{Q}$, $\mathcal{M}^{PT}_{GRPO}$ and $\mathcal{M}^{PT}_{HAR\text{-}GUI}$ are the post-training results on each benchmark for $\mathcal{M}_{Q}$, $\mathcal{M}_{GRPO}$ and $\mathcal{M}_{HAR\text{-}GUI}$, respectively. Compared with $\mathcal{M}_{GRPO}$, $\mathcal{M}^{*}_{GRPO}$ mandates that the agent focuses on history context of the episode in the instructions.

4.3 Ablation Study

We conduct an ablation study to validate our HAR framework. Starting from $\mathcal{M}_{warm\text{-}up}$, we tune GUI agents with the HAR framework to produce $\mathcal{M}_{HAR\text{-}GUI}$ and with vanilla GRPO[4] to yield $\mathcal{M}_{GRPO}$ and $\mathcal{M}^{*}_{GRPO}$. We then compare their screen grounding on ScreenSpot and episodic reasoning on AITW and Mind2Web. Further, we keep the same settings to conduct post-training on AITW and Mind2Web using $\mathcal{M}_{\mathcal{Q}}$, $\mathcal{M}_{GRPO}$ and $\mathcal{M}_{HAR\text{-}GUI}$ as initial checkpoints to verify the generalizability of our method.

Fig.3 illustrates that: ***(i)*** A comparison of $\mathcal{M}_{GRPO}$ and $\mathcal{M}_{warm\text{-}up}$ on ScreenSpot reveals that **training solely with episodic reasoning weakens the agent's grounding capability**, whereas our TMTS mitigates this degradation. ***(ii)*** Comparing $\mathcal{M}^{*}_{GRPO}$ and $\mathcal{M}_{GRPO}$, it is evident that a prompt-constrained focus on historical context can lead to performance degradation. We argue that **the agent's reasoning mode should not be constrained, but autonomously shaped and adapted during self-evolution**. ***(iii)*** On AITW and Mind2Web, $\mathcal{M}_{HAR\text{-}GUI}$ outperforms $\mathcal{M}_{GRPO}$, and in identical post-training settings, $\mathcal{M}^{PT}_{HAR\text{-}GUI}$ surpasses both $\mathcal{M}^{PT}_{GRPO}$ and $\mathcal{M}^{PT}_{Q}$. These results demonstrate the effectiveness of our HAR framework. Further, we observed that both $\mathcal{M}_{GRPO}$ and $\mathcal{M}^{*}_{GRPO}$ still exhibit weak short-term memory after RL training using the inference-format instruction. In contrast, the emerging history-aware and rigorous reasoning mode of $\mathcal{M}_{HAR\text{-}GUI}$ is driven by the HAR framework, primarily via **tailored guidance synthesis**, **cognitive correction in the reflection scenario**, and **MAR**.

[4]Without (*w/o*) reflection scenario, MAR, and TMTS.

Method	WC	S2W	TP	WS	QA	QAS	CQA
Llama 3.2-V (11B) (Meta AI 2024)	113.6	108.8	83.4	87.0	88.4	**91.6**	74.6
CogAgent (18B) (Hong et al. 2024)	136.2	115.0	**88.4**	63.1	85.3	74.6	65.1
UI-TARS-2B-SFT (Qin et al. 2025)	125.8	115.0	80.5	88.5	86.2	90.1	80.3
InternVL2 (8B) (Chen et al. 2024b)	140.6	115.2	86.7	89.7	84.2	89.2	82.4
InfiGUI-R1-3B (Liu et al. 2025b)	142.4	116.9	85.2	91.4	88.1	86.4	77.6
GUI-R1-3B (Luo et al. 2025)	141.7	117.5	84.9	92.2	88.3	87.2	79.0
Qwen2.5-VL-3B (Bai et al. 2025)	133.3	117.9	86.5	90.1	87.4	90.3	79.6
MP-GUI (8B) (Wang et al. 2025)	**151.0**	118.4	88.2	89.2	88.6	90.5	**84.3**
UI-TARS-1.5-7B (Qin et al. 2025)	147.4	118.7	87.1	**94.1**	**89.5**	91.2	82.1
HAR-GUI-3B	143.9	**119.1**	87.5	93.5	**89.5**	91.0	82.1

Table 8: Performance comparison on GUI understanding benchmarks. We employ CIDEr scores to assess WC and S2W, SQuAD F1 scores for QAS, CQA and WS, ROUGE-L scores for QA, and F1 values for TP.

5 Conclusion

We propose the HAR framework to enhance the reasoning of GUI agents via reflective training, especially equipping the agent with stable short-term memory for episodic reasoning. The framework consists mainly of constructing a reflective learning scenario, synthesizing tailored correction guidelines, and designing a hybrid RL reward function. Via HAR, we develop a native model HAR-GUI-3B, which integrates reliable performance to handle GUI-oriented tasks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62372408).This work was supported by Ant Group Research Fund.

References

Baechler, G.; Sunkara, S.; Wang, M.; Zubach, F.; Mansoor, H.; Etter, V.; Carbune, V.; Lin, J.; Chen, J.; and Sharma, A. 2024. ScreenAI: A Vision-Language Model for UI and Infographics Understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 3058–3068. ijcai.org.

Bai, H.; Zhou, Y.; Pan, J.; Cemri, M.; Suhr, A.; Levine, S.; and Kumar, A. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 12461–12495.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923.

Chen, M.; Li, Y.; Yang, Y.; Yu, S.; Lin, B.; and He, X. 2024a. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37: 589–631.

Chen, X.; Zhao, Z.; Chen, L.; Ji, J.; Zhang, D.; Luo, A.; Xiong, Y.; and Yu, K. 2021. WebSRC: A Dataset for Web-Based Structural Reading Comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4173–4185.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; YanTao, L.; Zhang, J.; and Wu, Z. 2024. SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9313–9332.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2Web: Towards a Generalist Agent for the Web. In *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*.

Gemini Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. Technical report, Google DeepMind.

Gou, B.; Wang, R.; Zheng, B.; Xie, Y.; Chang, C.; Shu, Y.; Sun, H.; and Su, Y. 2025. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; and Tang, J. 2024. CogAgent: A Visual Language Model for GUI Agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 14281–14290. IEEE.

Hsiao, Y.-C.; Zubach, F.; Baechler, G.; Carbune, V.; Lin, J.; Wang, M.; Sunkara, S.; Zhu, Y.; and Chen, J. 2022. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Li, K.; Meng, Z.; Lin, H.; Luo, Z.; Tian, Y.; Ma, J.; Huang, Z.; and Chua, T.-S. 2025. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*.

Li, Y.; Li, G.; He, L.; Zheng, J.; Li, H.; and Guan, Z. 2020. Widget Captioning: Generating Natural Language Description for Mobile User Interface Elements. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5495–5510.

Lin, K. Q.; Li, L.; Gao, D.; Yang, Z.; Wu, S.; Bai, Z.; Lei, S. W.; Wang, L.; and Shou, M. Z. 2025. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 19498–19508.

Liu, Y.; Li, P.; Wei, Z.; Xie, C.; Hu, X.; Xu, X.; Zhang, S.; Han, X.; Yang, H.; and Wu, F. 2025a. InfiGUIAgent: A Multimodal Generalist GUI Agent with Native Reasoning and Reflection. *arXiv preprint arXiv:2501.04575*.

Liu, Y.; Li, P.; Xie, C.; Hu, X.; Han, X.; Zhang, S.; Yang, H.; and Wu, F. 2025b. InfiGUI-R1: Advancing Multimodal GUI Agents from Reactive Actors to Deliberative Reasoners. arXiv:2504.14239.

Liu, Y.; Liu, J.; Shi, X.; Cheng, Q.; Huang, Y.; and Lu, W. 2024. Let's Learn Step by Step: Enhancing In-Context Learning Ability with Curriculum Learning. *arXiv preprint arXiv:2402.10738*.

Lu, Q.; Shao, W.; Liu, Z.; Meng, F.; Li, B.; Chen, B.; Huang, S.; Zhang, K.; Qiao, Y.; and Luo, P. 2024a. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*.

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

Lu, Z.; Chai, Y.; Guo, Y.; Yin, X.; Liu, L.; Wang, H.; Xiao, H.; Ren, S.; Xiong, G.; and Li, H. 2025. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*.

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*.

Meta AI. 2024. Llama 3. Technical report, Meta AI. Accessed: 2024-11-12.

Park, J.; Tang, P.; Das, S.; Appalaraju, S.; Singh, K. Y.; Manmatha, R.; and Ghadar, S. 2025. R-VLM: Region-Aware Vision Language Model for Precise GUI Grounding. arXiv:2507.05673.

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.

Schoop, E.; Zhou, X.; Li, G.; Chen, Z.; Hartmann, B.; and Li, Y. 2022. Predicting and explaining mobile ui tappability with vision modeling and saliency analysis. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–21.

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*.

Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366.

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.

Team, C.; Yue, Z.; Lin, Z.; Song, Y.; Wang, W.; Ren, S.; Gu, S.; Li, S.; Li, P.; Zhao, L.; Li, L.; Bao, K.; Tian, H.; Zhang, H.; Wang, G.; Zhu, D.; Cici; He, C.; Ye, B.; Shen, B.; Zhang, Z.; Jiang, Z.; Zheng, Z.; Song, Z.; Luo, Z.; Yu, Y.; Wang, Y.; Tian, Y.; Tu, Y.; Yan, Y.; Huang, Y.; Wang, X.; Xu, X.; Song, X.; Zhang, X.; Yong, X.; Zhang, X.; Deng, X.; Yang, W.; Ma, W.; Lv, W.; Zhuang, W.; Liu, W.; Deng, S.; Liu, S.; Chen, S.; Yu, S.; Liu, S.; Wang, S.; Ma, R.; Wang, Q.; Wang, P.; Chen, N.; Zhu, M.; Zhou, K.; Zhou, K.; Fang, K.; Shi, J.; Dong, J.; Xiao, J.; Xu, J.; Liu, H.; Xu, H.; Qu, H.; Zhao, H.; Lv, H.; Wang, G.; Zhang, D.; Zhang, D.; Zhang, D.; Ma, C.; Liu, C.; Cai, C.; and Xia, B. 2025. MiMo-VL Technical Report. arXiv:2506.03569.

Wang, B.; Li, G.; Zhou, X.; Chen, Z.; Grossman, T.; and Li, Y. 2021. Screen2words: Automatic mobile UI summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 498–510.

Wang, J.; Xu, H.; Jia, H.; Zhang, X.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024. Mobile-Agent-v2: Mobile Device Operation Assistant with Effective Navigation via Multi-Agent Collaboration. arXiv:2406.01014.

Wang, Z.; Chen, W.; Yang, L.; Zhou, S.; Zhao, S.; Zhan, H.; Jin, J.; Li, L.; Shao, Z.; and Bu, J. 2025. Mp-gui: Modality perception with mllms for gui understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 29711–29721.

Wu, P.; Ma, S.; Wang, B.; Yu, J.; Lu, L.; and Liu, Z. 2025a. GUI-Reflection: Empowering Multimodal GUI Models with Self-Reflection Behavior. *arXiv preprint arXiv:2506.08012*.

Wu, Q.; Liu, W.; Luan, J.; and Wang, B. 2025b. ReachAgent: Enhancing Mobile Agent via Page Reaching and Operation. arXiv:2502.02955.

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*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

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

Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Yue, Y.; Song, S.; and Huang, G. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? arXiv:2504.13837.

Zhang, C.; Yang, Z.; Liu, J.; Li, Y.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–20.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhou, Y.; Dai, S.; Wang, S.; Zhou, K.; Jia, Q.; and Xu, J. 2025. GUI-G1: Understanding R1-Zero-Like Training for Visual Grounding in GUI Agents. arXiv:2505.15810.

Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; Gao, Z.; Cui, E.; Wang, X.; Cao, Y.; Liu, Y.; Wei, X.; Zhang, H.; Wang, H.; Xu, W.; Li, H.; Wang, J.; Deng, N.; and Li., S. 2025. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. arXiv:2504.10479.

Zhuo, J.; Zhang, S.; Fang, X.; Duan, H.; Lin, D.; and Chen, K. 2024. ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 1950–1976. Miami, Florida, USA: Association for Computational Linguistics.