AnchorGUI: Asymmetric Memory for Dual-Scale Learning in GUI Navigation

Shengjie Jin[1], Zelong Sun[1], Hengbo Xu[1], Yanbiao Ma[1], and Zhiwu Lu[1⋆]

Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{jinshengjie,luzhiwu}@ruc.edu.cn

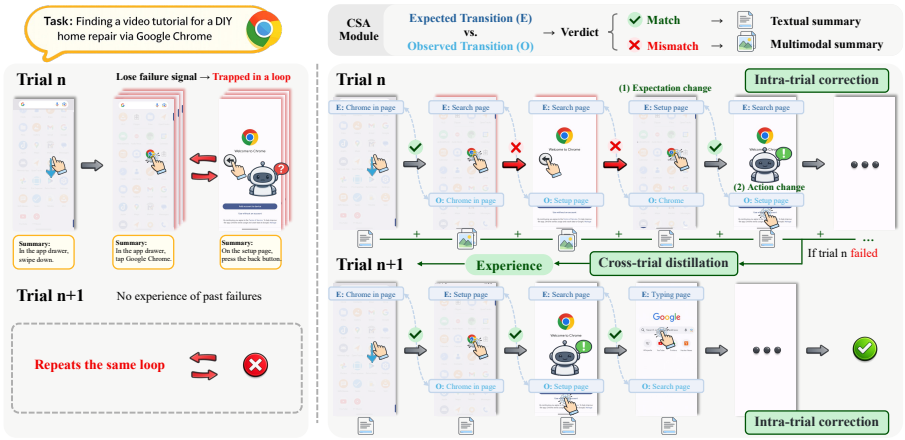
Abstract. Vision-Language Models (VLMs) enable autonomous GUI navigation, but agents still struggle to process and learn from dense, continuous visual histories. This bottleneck hinders both immediate error correction within a single episode (**intra-trial**) and experience distillation across multiple attempts (**cross-trial**). We trace these challenges to an empirical *informational asymmetry* in GUI navigation: while expected transitions can often be compressed into lightweight textual summaries, unexpected outcomes benefit from preserved screenshots as causal evidence for accurate diagnosis. Building on this insight, we propose **AnchorGUI**, a unified framework driven by the **Cognitive State Anchor (CSA)**. The CSA acts as a per-step primitive that actively compares expected and observed transitions, converting passive multimodal trajectories into explicit prediction-error signals. These signals orchestrate a **dual-scale learning mechanism** via an asymmetric memory. For *intra-trial correction*, a sliding window selectively retains visual evidence for detected mismatches, providing immediate, visually-grounded feedback. For *cross-trial distillation*, this asymmetric memory focuses the computationally expensive credit assignment search space on likely failure steps. Experiments across four benchmarks validate the effectiveness of our approach. On AndroidWorld, AnchorGUI achieves a 57.3% success rate with a 2.4× token reduction per step. Furthermore, cross-trial distillation reaches 69.2% success (+11.9% gain), significantly outperforming standard reflection methods while maintaining sub-linear context scaling.

Keywords: GUI Navigation · Vision-Language Models · Multimodal Agents · Asymmetric Memory

1 Introduction

Graphical User Interface (GUI) navigation requires autonomous agents to perceive visual interfaces and execute actions to accomplish user-specified goals [17, 27, 37]. This capability underpins a wide range of applications, from automated software testing to intelligent personal assistants [12, 40]. Recent advances in Vision-Language Models (VLMs) [2, 6, 8] have substantially expanded the potential of such agents, enabling them to navigate complex visual environments [24,

⋆ Corresponding author.



(a) Existing GUI Agents (b) AnchorGUI (Ours)

Fig. 1: Comparison of existing GUI agents and AnchorGUI. (a) Existing methods indiscriminately compress visual histories, discarding critical visual evidence of unexpected states. This causes agents to lose failure signals and repeat the same mistakes across trials. (b) **AnchorGUI** introduces Cognitive State Anchors (CSA) to explicitly compare expected versus observed transitions. By selectively retaining original screenshots for mismatched steps while compressing matched ones, it builds an asymmetric memory that enables efficient intra-trial correction and cross-trial distillation while avoiding multimodal token explosion.

38]. However, as illustrated in Figure 1, existing GUI agents face a significant bottleneck in **effectively processing and learning from dense, continuous visual interaction histories**, both within a single episode (**intra-trial**) and across multiple attempts (**cross-trial**).

Currently, most existing GUI agents prioritize **intra-trial** decision-making, predicting the next action based on accumulated interaction history [33, 34]. To mitigate multimodal token explosion, recent methods [14, 18, 29, 30] typically resort to aggressive truncation or the compression of visual histories into textual summaries. However, this indiscriminate compression treats successful and failed actions identically, discarding error signals essential for behavioral correction [13, 31] and preventing agents from recognizing and learning from their own mistakes. For instance, as shown in Figure 1, consider an agent searching for a video that becomes stuck on a "Welcome to Chrome" setup screen. If the visual evidence of this unexpected state is compressed into a generic representation indistinguishable from a normal search interface, the agent may repeatedly press "Back" and relaunch the app, trapped in a loop it cannot recognize or escape.

In practice, human users naturally overcome such deadlocks through **cross-trial** learning, distilling experience from past failures to ensure future success. While global reflection mechanisms have proven effective in text-based environments [20, 21, 35], applying them to visually grounded GUI tasks introduces a

severe *credit assignment bottleneck*. Unlike discrete text sequences, GUI trajectories consist of high-dimensional visual observations that often obscure the causal chain between actions and outcomes. Pinpointing the specific action responsible for a failure among dozens of visually similar steps is conceptually intractable and computationally expensive. Consequently, agents struggle to extract actionable insights from failures, even after multiple attempts.

These challenges in both intra-trial and cross-trial learning share a common root: the lack of a principled mechanism to distinguish which steps benefit from visual evidence and which can be abstracted. Our experiments reveal an empirical *informational asymmetry* in GUI navigation: the representational requirement of a step depends on whether the observed outcome aligns with the agent's expectations. When an action yields an anticipated visual transition (*e.g.*, a menu opening as expected), the state change can be effectively encapsulated by a lightweight textual summary. Conversely, when the observation deviates from expectation, this mismatch signals a breakdown in causal understanding. In such cases, textual abstraction may be insufficient to capture the complex dependency between the initial visual state and the failed action; the original screenshot can be retained as causal evidence for accurate diagnosis. This asymmetry can be used to address both challenges: for intra-trial learning, retaining visual evidence of mismatches enables immediate error correction; for cross-trial learning, it allows the agent to precisely locate failure-inducing actions, effectively bounding the search space for credit assignment.

Building on this insight, we propose **AnchorGUI**, a unified framework that addresses both intra-trial and cross-trial learning through a single architectural principle. At its core, AnchorGUI introduces the **Cognitive State Anchor (CSA)**, a per-step structured representation comprising an *Expected Transition*, an *Observed Transition*, and a binary *Verdict* indicating their alignment. The CSA anchors agent attention to specific states responsible for errors, preventing them from being lost in the stream of visual observations. Rather than passively recording trajectories, CSA actively compares pre-action expectations against post-action visual realities, converting dense multimodal streams into explicit prediction-error signals. These signals then orchestrate a **dual-scale learning mechanism**. For intra-trial correction, a sliding window maintains recent steps with their CSAs, selectively retaining visual observations for detected mismatches to provide immediate, visually-grounded feedback. For cross-trial distillation, this asymmetric memory aggregates across the entire episode, condensing expectation-consistent steps into text while preserving visual evidence primarily for detected mismatches. By exploiting informational asymmetry, AnchorGUI focuses the credit assignment search space on detected mismatches, enabling efficient experience extraction while drastically reducing multimodal token consumption at both temporal scales.

Our contributions are summarized as follows: **(1)** We identify the principle of **informational asymmetry** in GUI navigation, which provides a principled basis for selective visual retention based on expectation alignment. **(2)** We propose **AnchorGUI**, a unified framework with **CSA** module, which actively contrasts

expectations with observations to selectively retain visual evidence during compression. **(3)** We propose an asymmetric memory for both **intra-trial correction** and **cross-trial distillation**, which focuses credit assignment on detected mismatches. **(4)** Experiments on four benchmarks demonstrate AnchorGUI's effectiveness: it achieves a **57.3%** success rate on AndroidWorld with **2.4**× lower token cost (single-trial), and improves to **69.2%** (+**11.9%**) via asymmetric distillation (cross-trial), outperforming standard reflection methods.

2 Related Work

2.1 History Management in GUI Navigation

Long-horizon GUI navigation requires agents to condition decisions on extended interaction histories under dynamically evolving UI states [19, 32]. To manage growing context, existing methods adopt three strategies: (i) *context truncation* [18], which discards early steps entirely; (ii) *periodic summarization* [5, 30], which compresses trajectories into text; and (iii) *hierarchical planning* [1, 25, 26, 36], which decomposes tasks into subgoals to reduce the effective reasoning horizon. While these approaches mitigate token overhead, they share a common limitation: visual histories are primarily treated as passive context for next-step prediction, often discarding the fine-grained visual evidence required for diagnosing why a specific action failed. This loss of causal information severely hinders both immediate error correction and cross-trial knowledge transfer.

2.2 Step-Level Verification in GUI Agents

Recent work introduces step-level verification or feedback mechanisms to improve action reliability. GUI-Critic [28] employs a pre-execution critic to predict action outcomes, while MobileUse [10] and Mobile-Agent-v3 [36] integrates post-execution validation. These methods enable local error correction within a single trial. However, the verification signal is typically consumed immediately and then discarded, rarely being preserved or aggregated for future attempts. Consequently, an agent that encounters an unexpected pop-up in trial k has no mechanism to explicitly recall this experience in trial $k + 1$, forcing it to re-discover the same solution from scratch.

2.3 Cross-Trial Learning in Agents

Reflexion [20] demonstrates that verbal self-reflection enables effective cross-trial learning in text-based environments [21, 35]. However, directly applying this paradigm to GUI navigation introduces a multimodal reasoning challenge. Retaining full visual trajectories for cross-trial reasoning causes *context collapse*: the VLM must locate failure causes within an overwhelming volume of largely irrelevant screenshots, diluting attention and degrading credit assignment. Naïvely combining step-level verification with cross-trial reflection does not resolve this—reasoning still spans entire visual histories. The core challenge is to preserve only causal visual evidence while discarding redundant frames.

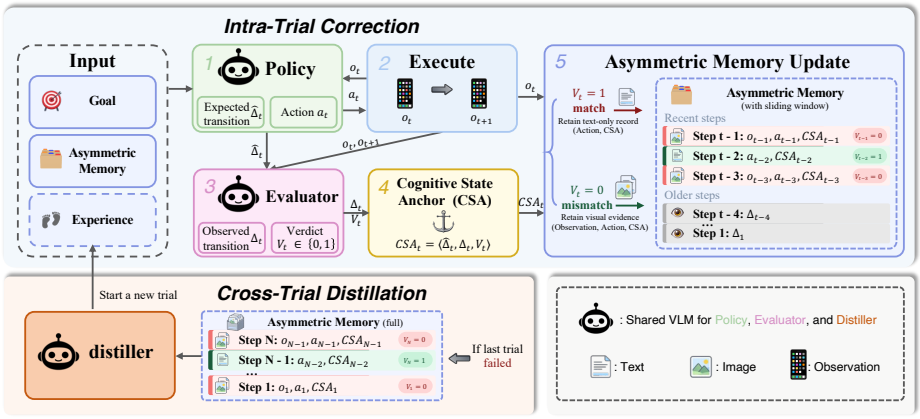


Fig. 2: Overview of AnchorGUI. At each step, the policy predicts an action together with an expected transition. After execution, the evaluator compares the expected transition with the observed GUI change and produces a Cognitive State Anchor (CSA). CSAs drive dual-scale learning: (*Top*) intra-trial correction via asymmetric memory updates, where mismatched steps retain visual observations while matched steps keep only text records within a sliding window, enabling efficient in-context correction; and (*Bottom*) cross-trial distillation that converts asymmetric memory into reusable experience when a trial fails, bounding the credit assignment search space.

3 Methodology

In this section, we present AnchorGUI, a unified framework designed to empower agents to effectively process and learn from dense, continuous visual interaction histories (see Figure 2). While existing agents often struggle with the inherent high-dimensionality of GUI trajectories, AnchorGUI introduces a principled mechanism to distinguish and retain critical visual evidence. We first formulate the visually-grounded decision process (Sec. 3.1). We then introduce the **Cognitive State Anchor (CSA)**, a structured representation that operationalizes the informational asymmetry between expected and observed transitions (Sec. 3.2). Finally, we detail how this design drives a dual-scale learning mechanism, enabling efficient intra-trial correction and cross-trial distillation without suffering from multimodal token explosion or intractable credit assignment (Sec. 3.3).

3.1 Problem Formulation

We define GUI navigation as a visually grounded Markov Decision Process (MDP) [23], formalized by the tuple

$$\mathcal{M} = \langle \mathcal{O}, \mathcal{A}, \mathcal{T}, G \rangle, \tag{1}$$

where $\mathcal{O}$ denotes the observation space, $\mathcal{A}$ the action space, $\mathcal{T}$ the environment transition function, and G a natural language goal describing the target task.

At each step t, the agent receives a visual observation

$$o_t \in \mathcal{O} = \mathbb{R}^{H \times W \times 3}, \tag{2}$$

corresponding to the current GUI screenshot. Let $\mathcal{H}_t$ denote the interaction history up to step t. Conditioned on the goal G and history $\mathcal{H}_t$, the policy π_θ predicts an action

$$a_t \sim \pi_\theta(a \mid G, o_t, \mathcal{H}_t), \quad a_t \in \mathcal{A}. \tag{3}$$

Following prior conventions [18], the action space $\mathcal{A}$ consists of predefined primitive GUI operations, including *click*, *swipe*, *type*, and *terminate*. Spatial actions are parameterized by continuous 2D coordinates (x, y) defined on the visual observation o_t. After executing action a_t, the environment transitions to the next observation according to

$$o_{t+1} \sim \mathcal{T}(o_t, a_t), \tag{4}$$

where $\mathcal{T}$ captures stochastic GUI dynamics such as rendering delays, animation transitions, or unexpected pop-ups. The episode terminates when the agent selects the *terminate* action or when a predefined maximum horizon T is reached.

3.2 Cognitive State Anchor: Bridging Expectation and Observation

To enable reliable credit assignment without indiscriminate compression, we introduce the *Cognitive State Anchor* (CSA). CSA acts as a principled interface between dense multimodal trajectories and reusable knowledge by explicitly modeling the alignment between the agent's internal belief and actual environment dynamics. It converts passive interaction histories into explicit prediction-error signals, forming the computational primitive for our framework.

AnchorGUI uses a single Vision-Language Model (VLM), parameterized by θ, with role-specific prompts for the *policy* (π_θ), *evaluator* (ϕ_θ), and *distiller* (ψ_θ). At step t, CSA is defined as:

$$CSA_t = \langle \hat{\Delta}_t, \Delta_t, V_t \rangle. \tag{5}$$

Expected Transition ($\hat{\Delta}_t$). Before execution, the policy role π_θ jointly predicts the spatial action and its anticipated visual effect:

$$(a_t, \hat{\Delta}_t) = \pi_\theta(G, o_t, \mathcal{H}_t), \tag{6}$$

where $\hat{\Delta}_t$ is formulated as a concise textual description (e.g., "*The 'Login' button will disappear, and the home feed will load*").

Observed Transition and Verdict (Δ_t, V_t). Upon reaching o_{t+1}, the evaluator role ϕ_θ compares the pre-action expectation against the post-action reality:

$$(\Delta_t, V_t) = \phi_\theta(\hat{\Delta}_t, o_t, o_{t+1}), \tag{7}$$

where Δ_t describes the actual UI change, and $V_t \in \{0, 1\}$ is a binary verdict indicating expectation alignment (1 for expectation-consistent, 0 for mismatch). To capture fine-grained visual differences, o_t and o_{t+1} are fed to ϕ_θ as a temporally ordered image sequence.

3.3 Dual-Scale Learning via Asymmetric Memory

Building upon these prediction-error signals, CSA drives the agent's learning cycle across two complementary temporal scales, exploiting informational asymmetry to optimize both intra-trial correction and cross-trial transfer.

Intra-Trial: Immediate Correction via Selective Visual Retention CSAs guide single-trial behavioral correction through **in-context learning** [3]. To support immediate error diagnosis under a bounded context, we apply an asymmetric retention strategy using a *sliding window* k.

For recent steps ($t-k \leq i < t$), we selectively retain visual observations based on the verdict V_i, preserving screenshots for detected expectation mismatches:

$$\mathcal{H}_{recent} = \bigoplus_{i=t-k}^{t-1} \begin{cases} \langle o_i, a_i, CSA_i \rangle & \text{if } V_i = 0, \\ \langle a_i, CSA_i \rangle & \text{if } V_i = 1, \end{cases} \tag{8}$$

where $\bigoplus$ denotes temporal concatenation. By explicitly including V_i in the prompt context, the policy π_θ learns to dynamically adjust its behavior: steps with $V_i = 0$ retain their visual context (o_i) to act as immediate, visually-grounded causal evidence for errors, while steps with $V_i = 1$ drop redundant images to save tokens, reinforcing successful local strategies purely via text.

For earlier steps ($i < t-k$), retaining images is computationally prohibitive. We thus aggressively compress them into a lightweight textual sequence of observed transitions: $\mathcal{H}_{old} = \bigoplus_{i=0}^{t-k-1} \langle \Delta_i \rangle$. The final effective context is constructed as $\mathcal{H}_t = \mathcal{H}_{old} \oplus \mathcal{H}_{recent}$. This asymmetric sliding window reduces multimodal context overhead from $\mathcal{O}(T)$ to $\mathcal{O}(|\{i \in [t-k, t) : V_i = 0\}|)$, supporting scalable reasoning across longer horizons.

Cross-Trial: Distillation and Bounded Credit Assignment Upon trial completion, the agent distills reusable strategies (e.g., handling specific pop-ups) for future attempts. Let $\tau^{(k)}$ denote the k-th trial, spanning T_k steps. Extending the selective retention strategy to the episode level, we construct a global **Asymmetric Memory** $\mathcal{H}_{asym}^{(k)}$ that aggregates the entire trial:

$$\mathcal{H}_{asym}^{(k)} = \bigoplus_{i=0}^{T_k} \begin{cases} \langle o_i, a_i, CSA_i \rangle & \text{if } V_i = 0, \\ \langle a_i, CSA_i \rangle & \text{if } V_i = 1. \end{cases} \tag{9}$$

As established, the *informational asymmetry* in GUI navigation motivates that successful actions ($V_i = 1$) can often be abstracted into text, whereas failed actions ($V_i = 0$) benefit from the original visual state (o_i) to resolve causal ambiguity. By localizing negative verdicts, this asymmetric memory explicitly bounds the temporal search space for credit assignment. It reduces the multimodal reasoning complexity for the distiller from the full trajectory length $\mathcal{O}(T)$ to the detected subset of mismatched steps $\mathcal{O}(|\{i : V_i = 0\}|)$, mitigating the

combinatorial ambiguity inherent in global reflection. The distiller role ψ_θ then processes this asymmetric memory to synthesize experience:

$$\mathcal{X}^{(k)} = \psi_\theta(\mathcal{H}^{(k)}_{asym}). \tag{10}$$

Rather than naïvely concatenating past reflections, $\mathcal{X}^{(k)}$ acts as an iteratively updated experience bank, formatted as a concise set of natural language strategic rules (e.g., "*If a pop-up blocks the screen, click the 'X' at the top right before proceeding*"). In the next trial $\tau^{(k+1)}$, the policy integrates this distilled experience to predict the next action, equipping π_θ with explicit strategies to avoid repeating past visual mistakes:

$$(a_t, \hat{\Delta}_t) = \pi_\theta(G, \mathcal{X}^{(k)}, o_t, \mathcal{H}_t). \tag{11}$$

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate AnchorGUI across four GUI navigation benchmarks. To assess intra-trial reasoning and immediate error correction, we employ three offline datasets: AITZ [39], Android-Control-High [11], and GUI-Odyssey [15]. To evaluate cross-trial distillation, we conduct multi-trial experiments on AndroidWorld [19], a dynamic benchmark that supports iterative agent-environment interactions across diverse tasks and applications.

Metrics. For AITZ, Android-Control-High, and GUI-Odyssey, we report Type Match (TM), measuring whether the predicted action category aligns with the ground truth, and Exact Match (EM), which additionally requires all action parameters to be correct. For AndroidWorld, we report Success Rate (SR). In cross-trial evaluations, SR@K denotes the cumulative success rate achieved by the K-th attempt across multiple interaction trials

Compared Methods. We compare against methods spanning two complementary dimensions. **Intra-trial baselines** operate in single-attempt settings, including: (1) *fine-tuning methods*: OS-Genesis [22], Aguvis [33], OdysseyAgent [15], UI-TARS [18], GUI-Critic [28], UGround [7], Aria-UI [34], and Agent-S2 [1]; (2) *zero-shot prompting methods*: ReAct [35], ReSum [30], Truncation [18], M3A [19], Self-Reflection [16], Mobile-Agent-v3 [36], and Chain-of-Memory [5]. **Cross-trial strategies** distill experience across multiple attempts; we evaluate Reflexion [20] and Episodic Memory [20], pairing each with different intra-trial backbones to isolate their individual contributions (Table 3). Detailed baseline descriptions are provided in the Supplementary Material.

Implementation Details. We adopt Qwen3-VL-8B [2] as the primary vision-language backbone model. We set the intra-trial sliding window size to k=2 and the maximum number of cross-trial attempts to K=3. All experiments are conducted with 3 different random seeds; we report the mean performance across runs. Complete hyperparameters and implementation details are provided in the Supplementary Material. The source code will be released upon publication.

Table 1: Performance comparison on offline GUI navigation benchmarks. All methods operate in single-trial setting. TM: Type Match; EM: Exact Match.

Method	Model	AITZ		Android-Control		GUI-Odyssey	
		TM (%)	EM (%)	TM (%)	EM (%)	TM (%)	EM (%)
Fine-tuning Methods							
OS-Genesis [22]	OS-Genesis-7B	20.0	8.45	65.9	44.4	11.7	3.63
Aguvis [33]	Aguvis-7B	35.7	19.0	65.6	54.2	26.7	13.5
OdysseyAgent [15]	OdysseyAgent	59.2	31.6	58.8	32.7	**90.8**	**73.7**
UI-TARS [18]	UI-TARS-7B	**71.7**	**55.3**	**68.5**	**60.8**	78.8	57.3
Zero-shot Methods							
GPT-4o [8]	GPT-4o	70.0	35.3	63.1	30.9	37.5	14.2
ReAct [35]	Qwen3-VL-8B	71.4	57.5	71.8	70.0	75.4	58.9
ReSum [30]	Qwen3-VL-8B	69.9	51.9	72.3	70.5	76.5	54.3
Truncation [18]	Qwen3-VL-8B	71.0	56.7	72.0	70.5	78.3	60.3
Self-Reflection [16]	Qwen3-VL-8B	73.9	58.0	71.9	70.3	80.1	**60.7**
AnchorGUI (Ours)	**Qwen3-VL-8B**	**75.2**	**59.1**	**74.6**	**73.0**	**80.6**	60.6

4.2 Main Results

Intra-Trial Correction on Offline Benchmarks. AnchorGUI demonstrates strong zero-shot performance in static environments by explicitly preserving visual causal evidence through asymmetric retention. As shown in Table 1, compared to full-history baselines such as ReAct, our method achieves higher performance on the AITZ dataset (+3.8% in TM and +1.6% in EM), suggesting that selective visual retention effectively avoids the attention dilution caused by blindly stacking historical steps. Furthermore, on the Android-Control benchmark, AnchorGUI surpasses text-compression baselines like ReSum by +2.3% in TM and +2.5% in EM. This trend implies that compressing post-action states into text discards the fine-grained visual evidence necessary to diagnose complex UI failures, whereas the comparative mechanism of the CSA retains this critical visual grounding. Compared to Truncation, which discards history indiscriminately, the verdict-driven retention of AnchorGUI presents a more principled strategy, as arbitrary truncation wastes tokens on successful transitions and risks losing critical causal failure frames. While the fine-tuned OdysseyAgent achieves higher scores on the GUI-Odyssey benchmark due to domain-specific training, AnchorGUI maintains competitive cross-domain adaptability as a zero-shot method without requiring task-specific weight updates.

Intra-Trial Performance on AndroidWorld. In the dynamic AndroidWorld environment, the single-attempt success rate of AnchorGUI surpasses both standard single-agent baselines and complex multi-agent frameworks. As shown in Table 2, AnchorGUI achieves an SR@1 of 57.3%, outperforming engineered systems such as Mobile-Agent-v3 (55.2%) and Chain-of-Memory (46.8%). This outcome indicates that a single binary verdict is sufficient to align expectations throughout the decision process, bypassing the coordination overhead of multi-agent pipelines or the multi-stage procedures required to maintain short-term and long-term mem-

Table 2: Single-trial performance on AndroidWorld. SR: Success Rate.

Method	Model	SR (%)
Fine-tuning Methods		
GUI-Critic [28]	GPT-4o + GUI-Critic-R1	29.4
UGround [7]	GPT-4o + UGround	32.8
Aguvis [33]	GPT-4o + Aguvis-7B	37.1
Aria-UI [34]	GPT-4o + Aria-UI	44.8
UI-TARS [18]	UI-TARS-72B-SFT	46.6
Agent-S2 [1]	Claude-3.7-Sonnet + UI-TARS-72B-DPO	**54.3**
Zero-shot Methods		
GPT-4o [8]	GPT-4o	34.5
AndroidGen [9]	GPT-4o	46.8
MobileUse [10]	Qwen2.5-VL-7B	21.6
MobileUse [10]	Qwen2.5-VL-32B	44.4
ReAct [35]	Qwen3-VL-8B	43.7
ReSum [30]	Qwen3-VL-8B	45.6
Truncation [18]	Qwen3-VL-8B	44.7
Self-Reflection [16]	Qwen3-VL-8B	47.7
M3A [19]	Qwen3-VL-8B	39.8
Mobile-Agent-v3 [36]	Qwen3-VL-8B	55.2
Chain-of-Memory [5]	Qwen3-VL-8B	46.8
AnchorGUI (Ours)	**Qwen3-VL-8B**	**57.3**

ory. From an efficiency perspective, AnchorGUI not only improves the SR@1 of the identically backed ReAct baseline by +13.6% but also effectively mitigates multimodal token explosion by reducing intra-trial consumption by a factor of 2.4 per step. This dual advantage suggests that the asymmetric memory mechanism successfully removes redundant images from successful steps, enabling a lightweight 8B model to maintain a high signal-to-noise ratio and outperform more complex architectures.

Cross-Trial Distillation Results. AnchorGUI exhibits consistent cross-trial improvement, which appears strongly correlated with its controlled treatment of the multimodal credit assignment problem. As detailed in Table 3, our approach reaches an SR@3 of 69.2% with an absolute learning gain of +11.9%, establishing a final performance margin of +14.0% over ReAct + Reflexion (55.2% SR@3). The significance of this gain is amplified when considering the principle of diminishing returns; achieving a +11.9% improvement from a high initial SR@1 of 57.3% represents a steeper learning curve than the +11.5% gain Reflexion achieves from a much lower 43.7% starting point. Furthermore, AnchorGUI accomplishes this with a cross-trial distillation cost of only 24.4k tokens, compared to roughly 60k tokens required by Reflexion-style baselines. The performance plateau of the baselines suggests a severe credit assignment bottleneck caused by unbounded search spaces that dilute attention across full trajectories. By filtering out successful frames, AnchorGUI biases the model toward failure-related signals, enabling efficient, low-cost rule extraction.

Table 3: Cross-trial learning performance and token efficiency on AndroidWorld. We compare system-level baselines with controlled ablations using different distillation strategies. **Intra** tokens: average per-step cost including policy generation and step-level verification (e.g., evaluator for AnchorGUI, self-reflection for baseline methods). **Cross** tokens: distillation cost per trial.

Method	Success Rate (%)				Tokens (Avg.)	
	SR@1	SR@2	SR@3	Δ (T3-T1)	Intra (/Step)	Cross (/Trial)
Baseline intra-trial + various cross-trial strategies						
ReAct + Episodic Memory	43.7	49.1	53.4	+9.74	91.4k	-
ReAct + Reflexion	43.7	**53.0**	55.2	**+11.5**	**32.7k**	**59.2k**
Self-Reflection + Episodic Memory	**47.7**	51.6	54.2	+6.50	98.5k	-
Self-Reflection + Reflexion	**47.7**	52.5	**56.8**	+9.13	38.1k	60.7k
AnchorGUI intra-trial + various cross-trial strategies						
Memoryless Retry	**57.3**	61.2	62.9	+5.51	13.5k	-
Episodic Memory	**57.3**	60.5	62.6	+5.29	78.9k	-
Reflexion	**57.3**	63.1	64.4	+7.02	13.6k	66.5k
Asymmetric Distillation (Ours)	**57.3**	**65.4**	**69.2**	**+11.9**	**13.7k**	**24.4k**

Table 4: Ablation study on AndroidWorld. Each row incrementally adds one component to validate its contribution. **Intra** tokens: per-step cost including policy and verification. **Cross** tokens: per-trial distillation cost. †: memoryless retry without cross-trial distillation.

Method / Component Added	Success Rate (%)				Tokens (Avg.)	
	SR@1	SR@2	SR@3	Δ (T3-T1)	Intra (/Step)	Cross (/Trial)
Intra-Trial Configurations						
(1) Baseline (ReAct)	43.7	50.9†	53.7†	**+10.0**	32.4k	-
(2) + Dual-Grained Memory with CSA	55.3	60.5†	**63.0**†	+7.69	17.9k	-
(3) + Asymmetric Window Retention	**57.3**	**61.2**†	62.9†	+5.51	**13.5k**	-
Cross-Trial Configurations						
(4) + Cross-Trial Distillation	**57.3**	63.8	66.1	+8.74	13.9k	65.3k
(5) + **Asymmetric Distillation (AnchorGUI)**	**57.3**	**65.4**	**69.2**	**+11.9**	**13.7k**	**24.4k**

4.3 Ablation Studies

Component Ablation Studies. The progressive ablation in Table 4 examines how the principle of informational asymmetry influences both performance and efficiency across two temporal scales. At the intra-trial scale (Rows 1 to 3), introducing the explicit CSA and asymmetric retention improves the SR@1 to 57.3% while simultaneously reducing the per-step token cost from 17.9k to 13.5k. At the cross-trial scale (Rows 4 to 5), asymmetric distillation (Row 5) attains a higher SR@3 (69.2%) than full visual distillation (Row 4, 66.1%) while consuming substantially fewer tokens. This comparison indicates that retaining the entire visual history paradoxically weakens credit assignment by obscuring causally relevant failure signals with a large volume of successful frames. Additionally, comparing

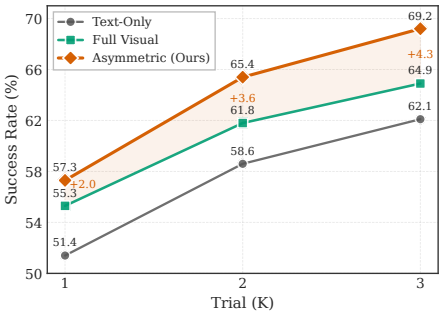


Fig. 3: Ablation on retention strategies. Cross-trial learning curves of different visual retention strategies.

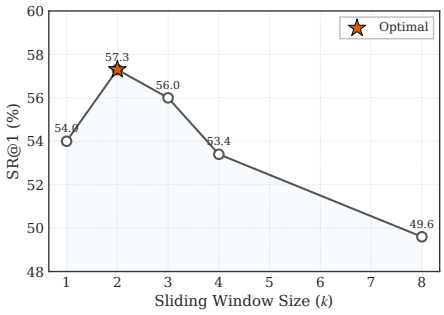


Fig. 4: Sliding window size impact. Single-trial performance across varying sliding window sizes k.

the memoryless retry gains helps isolate the role of environmental stochasticity; the smaller retry gain of Row 3 (+5.51%) compared to Row 1 (+10.0%) reflects the compressed room for stochastic improvement given Row 3's higher initial success rate. Consequently, the +11.9% learning gain achieved by asymmetric distillation (Row 5) implies that the model successfully extracts valid cross-trial knowledge rather than merely benefiting from environmental stochasticity.

Efficacy of Asymmetric Visual Retention. To validate our core hypothesis regarding informational asymmetry, we compare three unified retention strategies in Figure 3. The pure text retention strategy yields the lowest cross-trial performance with a 62.1% success rate at the third attempt, suggesting that discarding all visual history severs the causal chain necessary to diagnose complex UI failures. Conversely, the full visual retention strategy improves the success rate to 64.9% by preserving causal evidence, but it introduces significant visual noise by treating all steps equally. The asymmetric visual retention strategy achieves the steepest learning curve and the highest final success rate of 69.2%. This trend shows that preserving visual evidence for detected mismatches filters redundant successful transitions and anchors attention on failure signals.

Optimal Context Bounds in Intra-Trial Dynamics. Figure 4 illustrates the trade-off in single-trial performance as the sliding window size varies. Setting the window to $k = 2$ with asymmetric retention yields the optimal immediate correction capability with a 57.3% success rate. A minimal window of $k = 1$ drops performance to 54.0%, indicating that overly extreme compression sacrifices the necessary sequential context required to diagnose cascading UI errors. More importantly, expanding the window beyond $k = 3$ paradoxically degrades performance, bottoming out at 49.6% for $k = 8$. This decline reveals that even with asymmetric dropping, an excessively long temporal window accumulates historical noise and dilutes the policy attention.

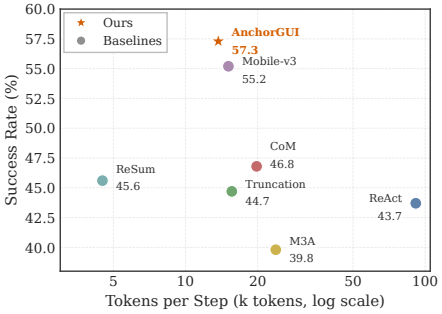


Fig. 5: Pareto frontier analysis. Task success rate versus average token cost per step across different methods.

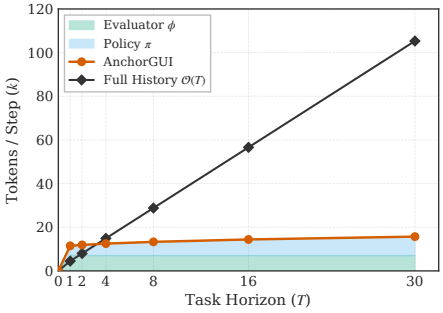


Fig. 6: Token scaling with task horizon. Token consumption scaling as the episode length T increases.

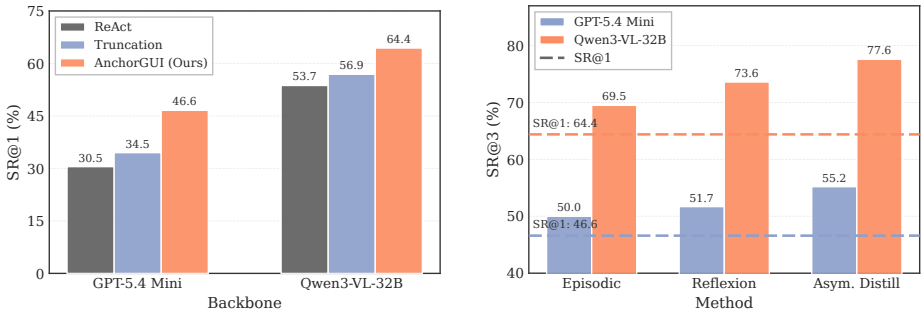
4.4 Efficiency and Scalability

Pareto Optimality in Performance and Efficiency. Figure 5 plots the Pareto frontier of task success rate versus total token cost per step, showing that AnchorGUI occupies the optimal top-left quadrant with a 57.3% success rate at an average cost of only 13.7k tokens. In contrast, standard uncompressed methods like ReAct consume a prohibitive 91.4k tokens for a much lower 43.7% success rate, and even engineered baselines like Mobile-Agent-v3 achieve 55.2% while consuming 15.1k tokens. This demonstrates that the localized 7k token overhead of the evaluator is more than offset by its ability to aggressively prune redundant visual histories, making the overall system highly cost-effective.

Token Scaling with Task Horizon. A central property of AnchorGUI is that its multimodal context complexity remains bounded by $\mathcal{O}(k)$, independent of episode length $\mathcal{O}(T)$. As shown in Figure 6, token consumption remains stable as interaction steps increase; a standard trajectory accumulation approach such as ReAct explodes linearly, reaching 105.3k tokens by step 30. In contrast, the policy token consumption of AnchorGUI remains remarkably stable, growing marginally from 4.5k tokens at step 1 to just 8.7k tokens at step 30. Even when adding the fixed 7k token overhead from the evaluator, the total cost at step 30 remains tightly constrained to 15.7k tokens. This sub-linear scaling implies that AnchorGUI effectively eliminates the multimodal token explosion bottleneck, which is a fundamental prerequisite for deploying autonomous agents in long-horizon navigation tasks.

4.5 Robustness and Generalization

Reliability of the Cognitive State Anchor. The efficacy of asymmetric memory depends on the accuracy of the evaluator's binary verdict. We manually annotated 535 interaction steps (see Supplementary Material for details). The zero-shot evaluator achieves 99.2% precision and 94.1% recall. From a system perspective,



(a) Intra-trial performance (SR@1) **(b)** Cross-trial performance (SR@3)

Fig. 7: Strong-backbone generalization. (a) Intra-trial performance (SR@1) and (b) cross-trial performance (SR@3) on AndroidWorld using GPT-5.4 Mini and Qwen3-VL-32B. All results are three-run averages on a fixed half subset.[1]

the extremely low false positive rate (0.56%) ensures that critical visual evidence of root failures is rarely discarded. The modest false negative rate (4.5%) incurs only negligible token overhead by occasionally retaining redundant successful frames, without affecting downstream causal reasoning. Overall, modern VLMs can reliably act as automated judges of expectation alignment, providing a solid foundation for asymmetric history orchestration.

Strong-Backbone Generalization. Figure 7 further evaluates GPT-5.4 Mini and Qwen3-VL-32B as strong backbones. AnchorGUI improves SR@1 over ReAct and Truncation for both, while asymmetric distillation yields the best SR@3, indicating that CSA-driven asymmetric memory remains beneficial for both stronger open-source and closed-source VLM settings.

5 Conclusion

In this paper, we propose **AnchorGUI** by identifying an empirical *informational asymmetry*, where expected transitions often compress into text while unexpected mismatches benefit from original screenshots. At its core, the Cognitive State Anchor (CSA) compares pre-action expectations with post-action visual realities to generate explicit prediction-error signals. These signals drive asymmetric visual memory for dual-scale learning: *intra-trial*, enabling immediate error correction while mitigating multimodal token explosion; *cross-trial*, focusing the computationally expensive credit assignment search space on detected mismatches, allowing agents to efficiently distill insights from past failures. Extensive experiments show that AnchorGUI significantly boosts success rates and reduces token consumption, offering a principled and efficient framework for multimodal GUI agents to process and learn from dense visual histories.

[1] For GPT-5.4 Mini evaluation, we follow UI-Ins [4] by using GPT-5.4 Mini as the planner and Qwen3-VL-8B for grounding localization.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (62376274, 62437002). Zhiwu Lu is the corresponding author.

References

1. Agashe, S., Wong, K., Tu, V., Yang, J., Li, A., Wang, X.E.: Agent S2: A compositional generalist-specialist framework for computer use agents. In: Conference on Language Modeling (COLM) (2025)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
4. Chen, L., Zhou, H., Cai, C., Zhang, J., Tong, P., Zhang, X., Kong, Q., Liu, C., Liu, Y., Wang, W., Wang, Y., Jin, Q., Hoi, S.: UI-Ins: Enhancing GUI grounding with multi-perspective instruction as reasoning. In: International Conference on Learning Representations (ICLR) (2026)
5. Gao, X., Hu, C., Chen, B., Li, T.: Chain-of-Memory: Enhancing GUI agents for cross-application navigation. arXiv preprint arXiv:2506.18158 (2025)
6. Gemini Team, Georgiev, P., Lei, V.I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
7. Gou, B., Wang, D.R., Zheng, B., Xie, Y., Chang, C., Shu, Y., Sun, H., Su, Y.: Navigating the digital world as humans do: Universal visual grounding for GUI agents. In: International Conference on Learning Representations (ICLR). pp. 30851–30883 (2025)
8. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: GPT-4o system card. arXiv preprint arXiv:2410.21276 (2024)
9. Lai, H., Gao, J., Liu, X., Xu, Y., Zhang, S., Dong, Y., Tang, J.: AndroidGen: Building an Android language agent under data scarcity. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 2727–2749. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.138
10. Li, N., Qu, X., Zhou, J., Wen, M., Du, K., Lou, X., Peng, Q., Wang, J., Zhang, W.: MobileUse: A hierarchical reflection-driven GUI agent for autonomous mobile operation. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 38, pp. 40361–40388. Curran Associates, Inc. (2025)
11. Li, W., Bishop, W., Li, A., Rawles, C., Campbell-Ajala, F., Tyamagundlu, D., Riva, O.: On the effects of data scale on UI control agents. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 37, pp. 92130–92154. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-2925

12. Liu, G., Zhao, P., Liang, Y., Liu, L., Guo, Y., Xiao, H., Lin, W., Chai, Y., Han, Y., Ren, S., Wang, H., Liang, X., Wang, W., Wu, T., Lu, Z., Chen, S., LiLinghao, Wang, H., Xiong, G., Liu, Y., Li, H.: LLM-Powered GUI agents in phone automation: Surveying progress and prospects. Transactions on Machine Learning Research (TMLR) (2025)
13. Liu, Y., Li, P., Xie, C., Hu, X., Han, X., Zhang, S., Yang, H., Wu, F.: InfiGUI-R1: Advancing multimodal GUI agents from reactive actors to deliberative reasoners. arXiv preprint arXiv:2504.14239 (2025)
14. Liu, Z., Li, J., Zhao, W.X., Gao, D., Li, Y., Wen, J.r.: PAL-UI: Planning with active look-back for vision-based GUI agents. arXiv preprint arXiv:2510.00413 (2025)
15. Lu, Q., Shao, W., Liu, Z., Du, L., Meng, F., Li, B., Chen, B., Huang, S., Zhang, K., Luo, P.: GUIOdyssey: A comprehensive dataset for cross-app GUI navigation on mobile devices. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22404–22414. IEEE (Oct 2025). https://doi.org/10.1109/ICCV51701.2025.02080
16. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-Refine: Iterative refinement with self-feedback. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 46534–46594. Curran Associates, Inc. (2023)
17. Nguyen, D., Chen, J., Wang, Y., Wu, G., Park, N., Hu, Z., Lyu, H., Wu, J., Aponte, R., Xia, Y., Li, X., Shi, J., Chen, H., Lai, V.D., Xie, Z., Kim, S., Zhang, R., Yu, T., Tanjim, M., Ahmed, N.K., Mathur, P., Yoon, S., Yao, L., Kveton, B., Kil, J., Nguyen, T.H., Bui, T., Zhou, T., Rossi, R.A., Dernoncourt, F.: GUI agents: A survey. In: Findings of the Association for Computational Linguistics (ACL). pp. 22522–22538. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.findings-acl.1158
18. Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al.: UI-TARS: Pioneering automated GUI interaction with native agents. arXiv preprint arXiv:2501.12326 (2025)
19. Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W., Li, W., Campbell-Ajala, F., Toyama, D., Berry, R., Tyamagundlu, D., Lillicrap, T., Riva, O.: AndroidWorld: A dynamic benchmarking environment for autonomous agents. In: International Conference on Learning Representations (ICLR). pp. 406–441 (2025)
20. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 8634–8652. Curran Associates, Inc. (2023)
21. Shridhar, M., Yuan, X., Côté, M.A., Bisk, Y., Trischler, A., Hausknecht, M.: ALFWorld: Aligning text and embodied environments for interactive learning. In: International Conference on Learning Representations (ICLR) (2021)
22. Sun, Q., Cheng, K., Ding, Z., Jin, C., Wang, Y., Xu, F., Wu, Z., Jia, C., Chen, L., Liu, Z., Kao, B., Li, G., He, J., Qiao, Y., Wu, Z.: OS-Genesis: Automating GUI agent trajectory construction via reverse task synthesis. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 5555–5579. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.277
23. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT Press, Cambridge, MA (1998)

24. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with Large Language Models. Transactions on Machine Learning Research (TMLR) (2024)
25. Wang, J., Xu, H., Jia, H., Zhang, X., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J.: Mobile-Agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 37, pp. 2686–2710. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-0088
26. Wang, J., Xu, H., Ye, J., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J.: Mobile-Agent: Autonomous multi-modal mobile device agent with visual perception. arXiv preprint arXiv:2401.16158 (2024)
27. Wang, S., Liu, W., Chen, J., Zhou, Y., Gan, W., Zeng, X., Che, Y., Yu, S., Hao, X., Shao, K., et al.: GUI agents with Foundation Models: A comprehensive survey. arXiv preprint arXiv:2411.04890 (2024)
28. Wanyan, Y., Zhang, X., Xu, H., Liu, H., Wang, J., Ye, J., Kou, Y., Yan, M., Huang, F., Yang, X., Dong, W., Xu, C.: Look before you leap: A GUI-Critic-R1 model for pre-operative error diagnosis in GUI automation. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 38, pp. 3907–3929. Curran Associates, Inc. (2025)
29. Wu, W., Zhou, K., Yuan, R., Yu, V., Wang, S., Hu, Z., Huang, B.: Auto-scaling continuous memory for GUI agent. arXiv preprint arXiv:2510.09038 (2025)
30. Wu, X., Li, K., Zhao, Y., Zhang, L., Ou, L., Yin, H., Zhang, Z., Yu, X., Zhang, D., Jiang, Y., et al.: ReSum: Unlocking long-horizon search intelligence via context summarization. arXiv preprint arXiv:2509.13313 (2025)
31. Xiao, H., Wang, G., Chai, Y., Lu, Z., Lin, W., He, H., Fan, L., Bian, L., Hu, R., Liu, L., Ren, S., Wen, Y., Chen, X., Zhou, A., Li, H.: UI-Genie: A self-improving approach for iteratively boosting MLLM-based mobile GUI agents. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 38, pp. 150376–150411. Curran Associates, Inc. (2025)
32. Xu, Y., Liu, X., Sun, X., Cheng, S., Yu, H., Lai, H., Zhang, S., Zhang, D., Tang, J., Dong, Y.: AndroidLab: Training and systematic benchmarking of Android autonomous agents. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 2144–2166. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.107
33. Xu, Y., Wang, Z., Wang, J., Lu, D., Xie, T., Saha, A., Sahoo, D., Yu, T., Xiong, C.: Aguvis: Unified pure vision agents for autonomous GUI interaction. In: International Conference on Machine Learning (ICML). pp. 69772–69805. PMLR (2025)
34. Yang, Y., Wang, Y., Li, D., Luo, Z., Chen, B., Huang, C., Li, J.: Aria-UI: Visual grounding for GUI instructions. In: Findings of the Association for Computational Linguistics (ACL). pp. 22418–22433. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.findings-acl.1152
35. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: ReAct: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR) (2023)
36. Ye, J., Zhang, X., Xu, H., Liu, H., Wang, J., Zhu, Z., Zheng, Z., Gao, F., Cao, J., Lu, Z., et al.: Mobile-Agent-v3: Fundamental agents for GUI automation. arXiv preprint arXiv:2508.15144 (2025)

37. Zhang, C., He, S., Qian, J., Li, B., Li, L., Qin, S., Kang, Y., Ma, M., Liu, G., Lin, Q., Rajmohan, S., Zhang, D., Zhang, Q.: Large Language Model-Brained GUI agents: A survey. Transactions on Machine Learning Research (TMLR) (2025)
38. Zhang, C., Yang, Z., Liu, J., Li, Y., Han, Y., Chen, X., Huang, Z., Fu, B., Yu, G.: AppAgent: Multimodal agents as smartphone users. In: ACM Conference on Human Factors in Computing Systems (CHI). pp. 1–20. ACM (Apr 2025). https://doi.org/10.1145/3706598.3713600
39. Zhang, J., Wu, J., Yihua, T., Liao, M., Xu, N., Xiao, X., Wei, Z., Tang, D.: Android in the zoo: Chain-of-Action-Thought for GUI agents. In: Findings of the Association for Computational Linguistics (EMNLP). pp. 12016–12031. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024). https://doi.org/10.18653/v1/2024.findings-emnlp.702
40. Zhao, K., Song, J., Sha, L., Shen, H., Chen, Z., Zhao, T., Liang, X., Yin, J.: GUI testing arena: A unified benchmark for advancing autonomous GUI testing agent. arXiv preprint arXiv:2412.18426 (2024)