MaDS: Long-Horizon GUI Automation via Synergizing Dual-Layer Memory and Multi-Round Debate

Pengchen Chen, Shi Chen†, Qiming Ye, Xinli Chen, Xinran Li, Wei Xiang†
International Design Institute of Zhejiang University, Zhejiang University
{22421365, shelleych, wxiang}@zju.edu.cn

Abstract

Automating Graphical User Interface (GUI) operations with Multimodal Large Language Models (MLLMs) is promising but remains bottlenecked in real-world long-horizon settings. Key challenges include ensuring precise grounding across diverse interfaces and handling irreversible errors in extended workflows. Current methods often struggle to distinguish targets in low Signal-to-Noise Ratio (SNR) environments and lack sufficient pre-execution verification to prevent error accumulation. To address this, we propose the Memory-augmented Debate System (MaDS). Specifically, MaDS combines: (1) a Dual-Layer Memory Module that integrates universal interaction priors with scenario-specific operational experience to mitigate grounding hallucinations; and (2) Multi-Round Debate that performs pre-execution verification, while transforming execution failures into retrievable Negative Warnings to reduce repeated errors. Additionally, we introduce MaDS-Benchmark, a benchmark for long-horizon mobile GUI tasks with process-oriented evaluation. Experiments show that MaDS achieves a 90.23% Task Success Rate on MaDS-Benchmark and strong performance on public benchmarks including AITW, AITZ, CAGUI, and GUIOdyssey.

1 Introduction

Automating Graphical User Interface (GUI) operations has emerged as a pivotal frontier for Multimodal Large Language Models (MLLMs) (Nguyen et al., 2025; Tang et al., 2025; Zhang et al., 2025a; Deng et al., 2024). While GUI agents have progressed in standard environments (Nguyen et al., 2025; Tang et al., 2025), scaling to real-world workflows presents a barrier, as mobile interfaces feature diverse elements, complex layouts, and deviations from standard structures (Gou et al., 2025; Rawles et al., 2023; Wu et al., 2024). Achieving reliability in this domain is a problem of visual perception that demands precise grounding among these irregularities (Gou et al., 2025; Lu et al., 2024), and a challenge of sequential decision-making under irreversibility (Tian et al., 2025). In long-horizon workflows, a single deviation can divert the agent onto an error trajectory from which recovery is impossible (Nguyen et al., 2025; Tang et al., 2025; Kong et al., 2025). Consequently, fulfilling these requirements faces two hurdles:

Challenge 1: Achieving grounding precision among diversity of interfaces. In real-world applications, agents must handle diverse elements and dynamic content where the Signal-to-Noise Ratio (SNR) varies significantly (Wu et al., 2025a; Gou et al., 2025). Since metadata-based methods are heavily dependent on the parsing of the underlying view hierarchies, they inherently lack universality when faced with custom-rendered interfaces or dynamic content where structural data are often encapsulated or obfuscated (Burns et al., 2021; Deng et al., 2023; Rawles et al., 2023; Venkatesh et al., 2023; Gur et al., 2024). Conversely, pure vision-based paradigms operate solely on raw pixel inputs, making them susceptible to the low SNR inherent in dense layouts (Lu et al., 2024; Xu et al., 2025b; Wang et al., 2025a; Baechler et al., 2024). This visual clutter often overwhelms the model's feature extraction capability, making it difficult to distinguish small targets from background noise and leading to grounding hallucinations.

Challenge 2: Overcoming irreversibility in long-horizon error recovery. Unlike static QA tasks, a single minor planning deviation in a GUI workflow can cascade into an error trajectory from which recovery is impossible (Yang et al., 2025a; Wu et al., 2025b). Current frameworks predominantly adopt a linear execution paradigm (Zhu et al., 2025; Agashe et al., 2025), leaving agents vulnerable to such error accumulation. Although some methods attempt post-hoc correction (Liu et al.,

†Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26158–26180

2024; Nayak et al., 2024; Putta et al., 2024), they operate on the assumption that errors are reversible. Since the damage is done the moment an action is executed, these remedial measures are inherently too late in real-world irreversible environments.

We propose the **M**emory-**a**ugmented **D**ebate **S**ystem (**MaDS**). To mitigate grounding hallucinations caused by low SNR in diverse interfaces (Challenge 1), MaDS incorporates a **Dual-Layer Memory Module** that synergizes universal semantic knowledge with specific operational experience, acting as contextual priors against visual noise. Crucially, to handle task irreversibility (Challenge 2), we implement **Multi-Round Debate** that enforces pre-emptive verification by subjecting plans to adversarial scrutiny and strict visual constraints before execution, while a continuous Reflection process converts failures into negative warnings to strictly prevent error recurrence.

The main contributions of this paper are summarized as follows, code and data are available for further research [1]:

- **MaDS:** A system driven by Dual-Layer Memory and Multi-Round Debate. In particular, MaDS emphasizes pre-execution verification and uses failure-derived Negative Warnings as retrievable constraints to reduce repeated errors in irreversible workflows.
- **MaDS-Benchmark:** A benchmark focusing on long-horizon tasks in real-world dynamic mobile scenarios. We also introduce a process-oriented evaluation method utilizing execution logs to attribute failures at the trace level.
- **Performance:** MaDS achieves a 90.23% Task Success Rate (TSR) on MaDS-Benchmark and demonstrates generalization on public datasets, attaining 94.74% on AITW (Rawles et al., 2023), 95.24% on AITZ (Zhang et al., 2024b), and 96.77% on CAGUI (Zhang et al., 2025b).

2 Related Work

2.1 Achieving Grounding Precision among Diversity of Interfaces

Achieving precise grounding across diverse real-world applications remains a primary hurdle (Yang et al., 2025b; Park et al., 2025; Wu et al., 2025a). Early approaches relied on scripts or metadata, but these lack universality as they are often restricted by custom rendering or dynamic content (Chen et al., 2025; Wen et al., 2024; Zhang et al., 2021). Consequently, the field has shifted towards MLLM-based architectures to process visual inputs directly (Hoscilowicz and Janicki, 2025; Ma et al., 2024; Xu et al., 2025b). However, models handling both high-level reasoning and pixel-level localization simultaneously face significant cognitive load (Xu et al., 2024; Zhang et al., 2024a). Diagnostic studies suggest that perceptual hallucinations (Tao et al., 2025; Wang et al., 2024b; Chen et al., 2024) are distinct from reasoning errors, advocating for systems that decouple perception from reasoning (Jia et al., 2025; Ni et al., 2025).

To support grounding in complex environments, recent research adopts dual-process memory architectures distinguishing between Episodic and Semantic Memory (Hu et al., 2025; Zhang et al., 2024e; Kim et al., 2023), as implemented in frameworks like Agent S (Agashe et al., 2025), HAR-GUI (Wang et al., 2025b) and MGA (Cheng et al., 2025). However, retrieval accuracy is often compromised by low SNR in dense, diverse interfaces (Hong and He, 2025; Agashe et al., 2025; Wang et al., 2025b). While approaches using graph-based retrieval or rigid filtering aim to address this (Shen et al., 2025; Guan et al., 2025), balancing scenario-specific precision with the recall of universal cross-app knowledge remains a critical optimization area to ensure reliable grounding.

Sustaining performance over long horizons also relies on effective memory maintenance (Wu et al., 2025d; Zhang et al., 2024e). Without proper management, data accumulation leads to information dilution, which degrades retrieval efficiency and consistency (Hong and He, 2025; Wu et al., 2025c). Current pruning methods predominantly utilize heuristic rules or embedding similarity (Hong and He, 2025; Zhang et al., 2024e). While recent works like MemoryField (Anonymous, 2025) and A-Mem (Xu et al., 2025a) explore autonomous curation, utilizing causal analysis to establish structured constraints for preventing repeated errors in irreversible tasks remains an open research direction.

2.2 Overcoming Irreversibility in Long-Horizon Error Recovery

In long-horizon tasks, error accumulation poses a severe threat to reliability. Multi-Agent Systems (MAS) like Mobile-Agent (Ye et al., 2025), GAIR (Wei et al., 2025) and COLA (Zhao et al., 2025) attempt to mitigate this by separating planning

[1] https://github.com/PcCin37/MaDS

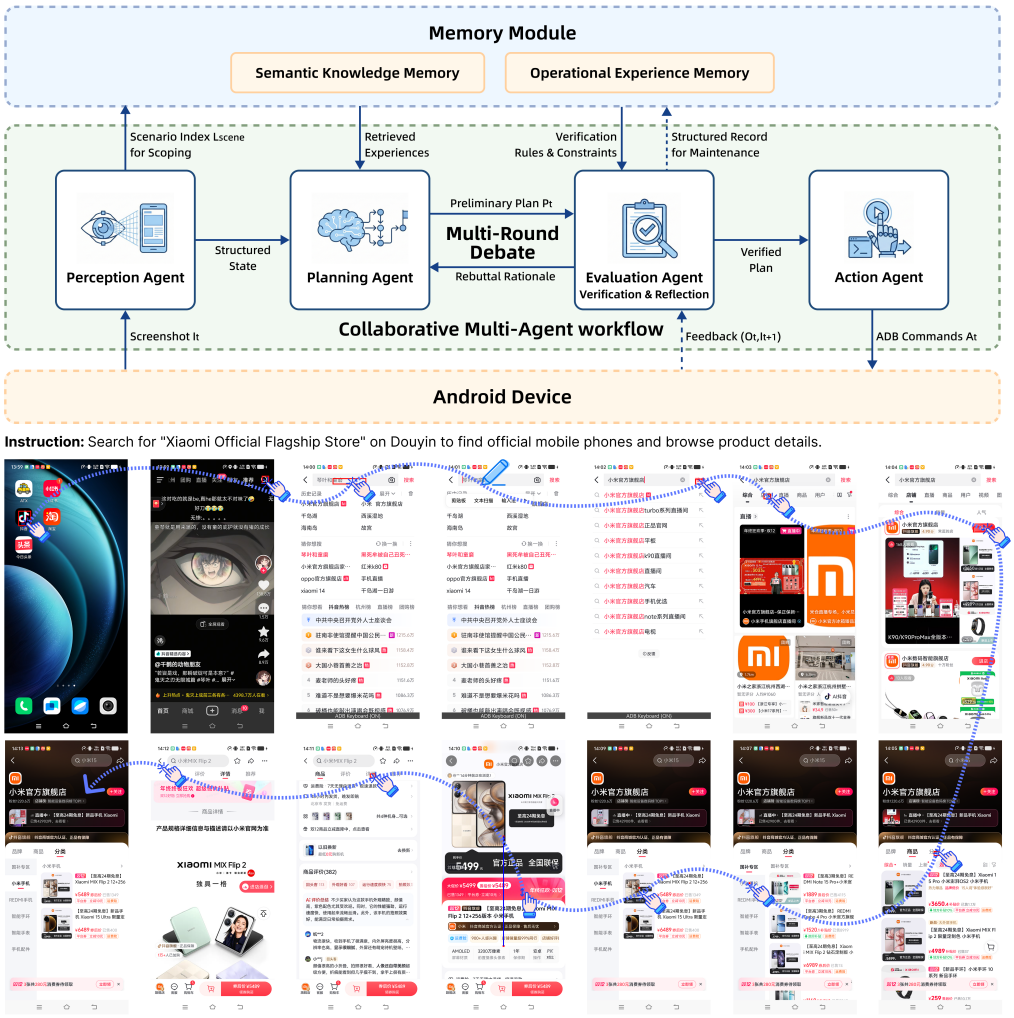


Figure 1: The Overall Architecture of **MaDS**.

from execution. However, hierarchical structures often prioritize upstream instructions (Cemri et al., 2025), allowing initial planning deviations to propagate downstream (Xinjie et al., 2025; Yang et al., 2025a). Although frameworks like COLA (Zhao et al., 2025) and AgentCPM-GUI (Zhang et al., 2025b) introduce reviewer roles to challenge reasoning paths (Roh et al., 2025), these function as post-hoc validators. In irreversible environments, such remedial correction is often too late, and there remains a need for pre-emptive verification to check logical consistency before physical operation (Wanyan et al., 2025; Zhang et al., 2024d).

3 System

The overall architecture of MaDS is illustrated in Figure 1.

3.1 Memory Module

The core objective of the Memory Module is to equip the agent with an evolutionary memory for long-horizon GUI interaction. In dynamic mobile environments, relying only on pre-trained knowledge is insufficient. An effective system must leverage broadly applicable interaction priors for generalized reasoning, while continuously accumulating scenario-specific experience to navigate complex layouts and reduce repeated mistakes in irreversible workflows. To achieve this, MaDS implements a Memory Module consisting of a **Dual-Layer Storage Structure** (Section 3.1.1), which separates Semantic Knowledge Memory from Operational Experience Memory, and a **Retrieval-and-Maintenance** mechanism (Section 3.1.2), which enables the memory to evolve with the agent's interactions.

3.1.1 Dual-Layer Storage Structure

Semantic Knowledge Memory encapsulates broadly applicable interaction norms and static facts derived from Android Material Design guidelines (Developers, 2024). We organize these as **Universal Priors**, covering action definitions, common

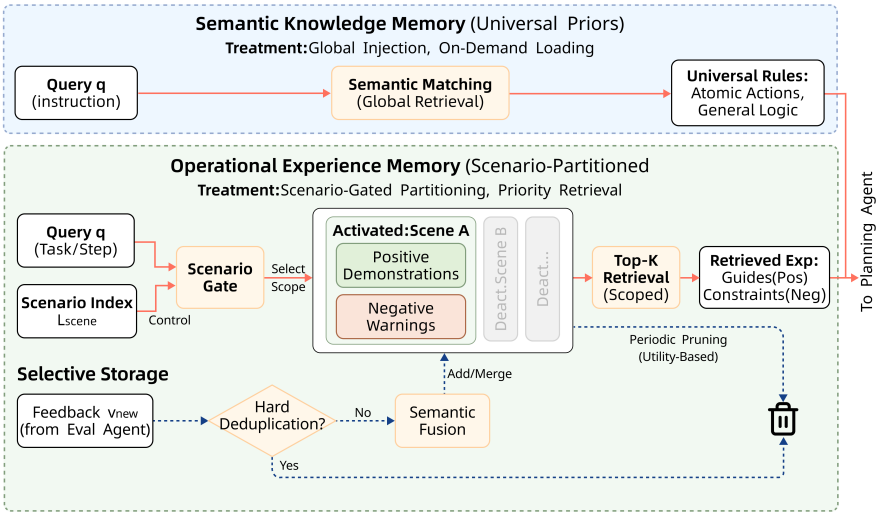


Figure 2: Structure and Mechanism of the Memory Module.

visual semantics, and general interface transition rules. This layer is lightweight, pre-installing 43 context-independent facts to provide probabilistic priors for reasoning, helping the agent adhere to basic interaction logic even in unseen apps within the evaluated mobile setting. The detailed content is listed in Table 6.

Operational Experience Memory stores evolving experiences accumulated during dynamic interactions. Unlike metadata-based methods that rely on fragile DOM structures (Burns et al., 2021; Deng et al., 2023), and unlike memory designs that primarily reuse successful trajectories, MaDS introduces three noise-resistant techniques to handle visual diversity and task irreversibility:

- **Soft Partitioning**: Instead of rigid database sharding, we use the Task Scenario L_{scene} as a soft partitioning index. The Perception Agent abstracts the current screen I_t into a semantic scenario. It clusters related experiences to maximize the SNR. This structure effectively supports knowledge transfer across similar apps.
- **Abstract Process Signature** S_{sig}: To decouple memory from pixel-level noise, we use S_{sig} to record the **Action Intent**. Unlike raw coordinates which are susceptible to resolution changes or dynamic layout shifts, S_{sig} records the action type and relative semantic target.
- **Dialectical Learning**: To counter the irreversibility of errors, the memory stores not only verified **Positive Demonstrations** but also failed operations with causal analysis as **Negative Warnings**. These warnings serve as hard constraints to prune the search space, preventing the agent from re-entering error trajectories.

Formally, memory nodes are stored as:

$$v = (L_{scene}, D_{task}, S_{sig}, \mathcal{F}_{out}, Label_{suc}, Reason)$$

In this representation, L_{scene} and S_{sig} allow experience to be retrieved at the appropriate semantic granularity, $\mathcal{F}_{out}$ is a hash summary of the execution result used for deduplication. $Label_{suc}$ and $Reason$ distinguish reusable positive demonstrations from failure-derived constraints.

3.1.2 Retrieval and Maintenance

MaDS employs a dynamic mechanism to orchestrate retrieval and maintenance, enabling evolutionary learning.

Retrieval Strategy. To combine general priors with scenario-specific experience under long-horizon decision making, the retrieval process is formulated as follows (see Appendix A.2 for parameter settings):

$$R(q, \mathcal{M}) = \text{TopK}(q, \mathcal{M}_{sem}) \cup \text{TopK}\big(q, \text{Scope}(\mathcal{M}_{exp}, L_{scene}|\tau)\big)$$

The query q encodes the global task D_{task} during initial planning, but shifts to the specific step instruction q_{step} during the Multi-Round Debate to retrieve fine-grained constraints.

- **Instruction-Driven Semantic Retrieval.** For Semantic Knowledge Memory $\mathcal{M}_{sem}$, the system adopts an on-demand retrieval. The algorithm performs semantic matching between q and universal priors to recall Top-K facts relevant to the immediate interaction norms.

- **Scenario-Prioritized Experience Retrieval.** For Operational Experience Memory $\mathcal{M}_{exp}$, the system implements a Priority-Based Strategy to increase precision. The function Scope$(\cdot)$ restricts the retrieval to the partition defined by L_{scene}, significantly increasing the effective SNR. Crucially, a dynamic fallback allows the scope to expand to the global pool if the similarity score falls below τ_{recall}, balancing specificity with recall. This design aims to balance precision and recall: scenario-prioritized retrieval increases specificity in dense, app-specific interfaces, while global fallback prevents the system from becoming overly narrow when local experience is insufficient.

Maintenance. To ensure long-term efficiency, the Maintenance adopts a Selective Storage Strategy (Algorithm 1). New experiences v_{new} generated from the Reflection undergo: (1) **Hard Deduplication** using $\mathcal{F}_{out}$ to discard redundant records, (2) **Semantic Fusion** to merge new insights with existing nodes, and (3) **Utility-Based Pruning** to remove stale nodes while preserving Negative Warnings, ensuring safety constraints are never forgotten.

3.2 Collaborative Multi-Agent Architecture

The objective of the Collaborative Multi-Agent Architecture is to address the irreversibility of long-horizon tasks by shifting from linear execution to a dual-phase process of **Pre-Emptive Verification** and **Continuous Evolution**. In environments where a single deviation can lead to unrecoverable failure, the system must examine candidate plans before execution and extract causal feedback after interaction. To achieve this, MaDS orchestrates four agents into a closed-loop system, using **Multi-Round Debate** (Section 3.2.1) as a pre-execution verification mechanism to intercept risky actions, and updating memory via **Reflection** (Section 3.2.2) to reduce the recurrence of error trajectories.

3.2.1 Multi-Round Debate

Directly executing retrieved plans carries risks: historical experience may contain noise, and the Planning Agent may over-rely on outdated spatial assumptions while ignoring current layout shifts. To prevent the agent from committing to such error trajectories, we design **Multi-Round Debate** (Figure 3). Unlike approaches that mainly provide corrective feedback after risky plans are formed or executed, our debate mechanism serves as a pre-execution verification process that checks reasoning against current visual evidence and retrieved memory constraints before physical action is taken.

The workflow operates as follows:

1. Preliminary Planning: The Planning Agent generates a plan P_t based on the global task and retrieved experiences.

2. Adversarial Scrutiny: While the Planner focuses on the global context (q_{task}), the Evaluation Agent performs a step-specific retrieval using the current instruction q_{step} to recall precise grounding constraints and Negative Warnings from the Memory Module. This Cross-Check ensures that high-level intent does not violate low-level physical constraints. In particular, retrieved Negative Warnings act as failure-derived constraints, enabling the Evaluation Agent to reject actions that resemble previously observed error trajectories rather than only scoring them by apparent semantic relevance.

3. Structured Confidence Scoring: Instead of free-form feedback, the Evaluation Agent outputs a structured assessment based on criteria (see Appendix A.1 for parameter settings):

$$C_{final} = w_1 C_{rule} + w_2 C_{ground} + w_3 C_{logic}$$

- **Rule Compliance** $C_{rule} \in \{0, 1\}$: A binary indicator. $C_{rule} = 0$ if the proposed action violates any atomic interaction norms defined in the Semantic Knowledge Memory or Negative Warnings from Operational Experience Memory; otherwise $C_{rule} = 1$.
- **Visual Grounding** $C_{ground} \in \{0, 1\}$: A binary constraint validating Physical Existence. The Evaluation Agent performs a QA check on the screenshot I_t: "Is the target element [Description] visible?". We set $C_{ground} = 1$ if the element is confirmed visible, and $C_{ground} = 0$ otherwise. We emphasize that C_{ground} is not provided by an external oracle or a dedicated detector. Instead, it serves as a conservative visibility check performed by the Evaluation Agent. The evaluator returns a positive grounding signal only when the target is clearly and unambiguously visible in the screenshot.
- **Logical Consistency** $C_{logic} \in [0, 1]$: A scalar score generated by the Evaluation Agent via prompting. The MLLMs evaluate the semantic relevance between the planned action A_t

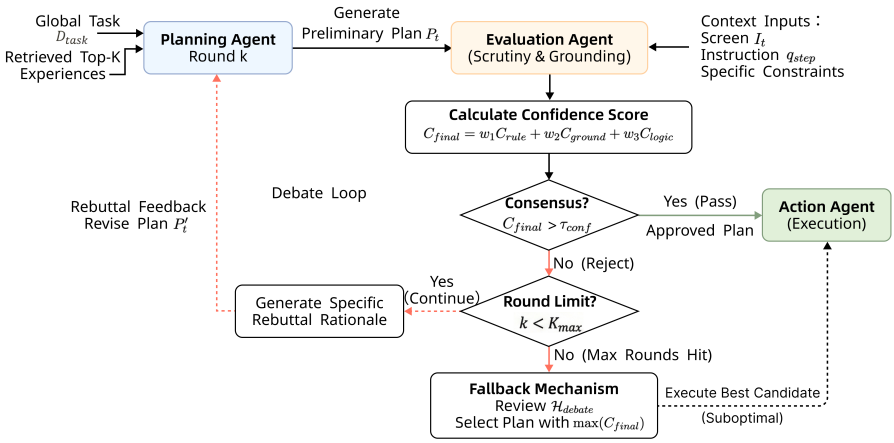


Figure 3: Workflow of the Multi-Round Debate.

and the step instruction q_{step} on a scale of 0 to 10, which is then normalized to [0, 1]. This measures whether the action is contextually optimal assuming the target exists.

4. Controllable Termination & Fallback: To address the unpredictability of LLM interactions, we introduce the following procedure. If C_{final} is below the threshold τ_{conf}, the Evaluation Agent provides specific rebuttal to trigger replanning. To prevent infinite cognitive cycling, we enforce a limit K_{max}. If no consensus is reached, the system executes the plan with the highest recorded confidence, ensuring the task proceeds rather than stalling.

3.2.2 Reflection

The goal of Reflection is to close the loop between execution and planning by transforming transient interactions into persistent memory. Rather than merely summarizing actions, MaDS uses diagnostic reflection to convert execution outcomes into reusable experience for future decision making. The Evaluation Agent receives the Visual Evidence Pair (pre- and post-action screenshots) together with the Operational Intent to analyze the execution outcome.

By parsing semantic changes, the agent adjudicates the outcome ($Label_{suc}$) and attributes the cause ($Reason$). This output is encapsulated into an experience node v_t and backpropagated to the Memory Module. Crucially, this directly addresses Challenge 2: failures are converted into **Negative Warnings**, which are subsequently retrieved during the Debate phase to serve as hard constraints, preventing the agent from re-entering error trajectories. At the same time, the quality of reflected experience remains bounded by the underlying model's ability to correctly interpret visual changes and execution outcomes, which may still be imperfect in highly ambiguous cases.

4 MaDS-Benchmark

We propose **MaDS-Benchmark**, comprising **271** real-world tasks spanning **11** top-tier applications across global and Chinese markets. Crucially, the average task length reaches **15.5** steps, substantially exceeding that of mainstream GUI-agent benchmarks (Table 7). Rather than relying solely on outcome-based evaluation, MaDS-Benchmark adopts a process-oriented protocol that helps localize failures to specific stages such as perception, retrieval, planning, and execution.

4.1 Task Design

To ensure broad coverage of realistic mobile workflows, we established three representative scenarios: **Content**, **E-commerce**, and **Services**. Targeting these scenarios, we selected widely used applications with over 100 million Daily Active Users (DAU), as detailed in Table 1.

These scenarios present specific challenges inherent to real-world workflows:

Dynamic Unstructured Layouts. Tasks take place in apps featuring live windows, infinite feeds, and transient visual noise (e.g., Danmu/bullet comments). The system must demonstrate the ability to filter noise and capture targets in dynamic pages filled with random pop-ups.

Long-Horizon and Strict Constraints. Tasks span multiple pages with deep navigation hierarchies. Agents must make complex sequential decisions while satisfying strict parameter constraints, such as specific dates, passenger counts, or price ranges.

To improve benchmark validity, we adopted a collaborative construction and verification proto-

Table 1: Core Scenarios and Covered Applications in MaDS-Benchmark.

Scenarios	Applications (China)	Applications (Global)
Content	Douyin, Xiaohongshu, Weibo, Toutiao	YouTube, Instagram
E-commerce	Douyin, Taobao, JD.com	Amazon
Services	Douyin, Meituan, Ctrip	Booking

col involving 3 master's students, 2 undergraduate students, 2 associate professors, and consultations with 10 industry practitioners from product, UI/UX, data, and engineering backgrounds. Each task underwent cross-review by 5-8 participants, and ambiguous, infeasible, or interactionally inconsistent tasks were removed. Only tasks verified to be fully executable were retained in the final set of 271 tasks.

4.2 Process-Oriented Evaluation

To enable a more granular analysis of agent behavior in dynamic environments, we adopt a Process-Oriented Evaluation protocol. Unlike evaluations that record only final success or failure, our framework preserves a step-level execution trace throughout task completion. During execution, the framework records a structured log (*history.jsonl*) capturing the Visual Evidence Pair, Operational Intent, and Causal Analysis at each step (data format shown in Appendix D.2). This enables detailed evaluation by linking final outcomes to specific breakdowns in perception, memory retrieval, planning, or logical reasoning.

Beyond improving diagnostic granularity, this protocol also reduces ambiguity in benchmark interpretation, since failures can be traced to specific intermediate breakdowns rather than being treated as undifferentiated end-task failures.

5 Experiments

In this section, we evaluate MaDS through comparative experiments against strong baselines and ablation studies on its components. Our experimental design examines the system from two perspectives: (1) **Comparative Evaluation:** We compare MaDS against foundation MLLMs and specialized agent frameworks to assess its effectiveness on long-horizon, cross-page, and dynamically complex workflows; (2) **Ablation Analysis:** We conduct ablation studies to isolate the contributions of the Dual-Layer Memory Module and the Multi-Round Debate.

5.1 Experimental Setup

To evaluate long-horizon planning and generalization, we utilize the proposed MaDS-Benchmark alongside three public datasets: AITW (Rawles et al., 2023), AITZ (Zhang et al., 2024b), CAGUI (Zhang et al., 2025b), and GUIOdyssey (Lu et al., 2025) as an additional external benchmark. We compare MaDS against two baseline groups: (1) **Foundation MLLMs**, including GPT-5 (OpenAI, 2025), Gemini-3-Pro (DeepMind, 2025), Qwen-2.5-VL (Qwen, 2025), Doubao-1.5-UI-TARS (Seed, 2025), and Claude-3.5-Sonnet (Anthropic, 2024), to assess intrinsic end-to-end capabilities; and (2) **Specialized Agent Frameworks**, including AgentCPM-GUI (Zhang et al., 2025b), AutoGLM (Liu et al., 2024), and AndroidArenaAgent (Xing et al., 2024), to evaluate architectural effects beyond the base models themselves.

Performance is measured using **T**ask **S**uccess **R**ate (**TSR**) for final outcomes, alongside **S**tep-level **P**lanning **S**uccess **R**ate (**SPSR**) and **S**tep-level **S**uccess **R**ate (**SSR**) to diagnose planning and grounding errors. Detailed definitions of datasets, baselines, and metrics are provided in Appendix D and E.

5.2 Main Results

5.2.1 Foundation MLLMs

We first evaluate several foundation MLLMs to assess their intrinsic end-to-end capability in GUI automation. As shown in Table 2, all evaluated foundation models obtain a TSR of 0.00% on MaDS-Benchmark, while achieving non-zero success rates on existing public benchmarks including AITW, AITZ, and CAGUI. These results suggest that foundation MLLMs are not uniformly incapable of GUI automation. Instead, their failures on MaDS-Benchmark reflect the difficulty of maintaining reliable performance over substantially longer task horizons.

A closer look at the step-level metrics reveals different failure patterns. General-purpose MLLMs such as GPT-5 and Claude-3.5-Sonnet show relatively strong intent understanding, achieving high

Table 2: Performance of Foundation MLLMs on public benchmarks and MaDS-Benchmark. The metrics are reported in the format of **TSR/SPSR/SSR** (%).

Foundation MLLMs	**AITW** (TSR / SPSR / SSR)	**AITZ** (TSR / SPSR / SSR)	**CAGUI** (TSR / SPSR / SSR)	**MaDS-Benchmark** (TSR / SPSR / SSR)
GPT-5	2.70 / 91.86 / 46.33	0.00 / 91.26 / 25.97	0.00 / 93.70 / 19.53	0.00 / **94.21** / 18.01
Claude-3.5-Sonnet	35.13 / 94.99 / 72.51	42.86 / 96.21 / 78.21	32.26 / 93.70 / 78.23	0.00 / 90.96 / 77.76
Doubao-1.5-UI-TARS	37.84 / 93.23 / 90.43	47.62 / 76.79 / 87.32	19.35 / 54.33 / 71.93	0.00 / 43.51 / **87.71**
Qwen-2.5-VL-7B	37.84 / 90.98 / 90.33	52.38 / 88.88 / 89.65	38.70 / 77.98 / 83.20	0.00 / 73.59 / 86.39
Gemini-3-Pro	5.41 / 88.60 / 47.67	4.76 / 81.91 / 65.58	0.00 / 80.64 / 53.15	0.00 / 79.61 / 49.14

Table 3: Comparison with SOTA GUI agents on public benchmarks and MaDS-Benchmark. The metrics are reported in the format of **TSR/SPSR/SSR** (%).

Agent	**AITW** (TSR / SPSR / SSR)	**AITZ** (TSR / SPSR / SSR)	**CAGUI** (TSR / SPSR / SSR)	**MaDS-Benchmark** (TSR / SPSR / SSR)
AndroidArenaAgent	81.08 / 91.86 / 96.27	71.43 / 85.42 / 87.64	64.52 / 78.49 / 87.39	0.00 / 39.91 / 79.24
AgentCPM-GUI	86.49 / 95.56 / 96.69	80.95 / 94.08 / 94.24	77.42 / 94.37 / 95.89	42.86 / 64.10 / 90.72
AutoGLM	83.78 / 90.46 / 86.16	90.48 / 95.17 / 97.15	90.32 / 96.15 / 98.44	71.43 / 86.69 / 95.48
MaDS (Ours)	**94.74 / 96.41 / 97.03**	**95.24 / 96.83 / 97.49**	**96.77 / 98.24 / 99.37**	**90.23 / 95.83 / 98.34**

Note: We also evaluated other agent frameworks, like Mobile-Agent (Wang et al., 2024a) and Agent S (Agashe et al., 2025). However, they are excluded from this table as they failed to successfully execute the initial step across the MaDS-Benchmarks.

Table 4: Ablation study on the contribution of each component. The metrics are reported as **TSR/SPSR/SSR/RSR** (%).

Sem.	**Exp.**	**Deb.**	**TSR / SPSR / SSR / RSR**
×	×	✓	5.56 / 57.86 / 66.35 / 26.71
✓	×	✓	19.44 / 63.20 / 66.89 / 12.23
×	✓	✓	72.22 / 82.94 / 88.61 / 40.54
✓	✓	×	22.22 / 65.25 / 67.24 / 36.20
✓	✓	✓	91.67 / 88.25 / 91.25 / 85.71

SPSR on public benchmarks. For example, GPT-5 reaches over 91% SPSR across AITW, AITZ, and CAGUI, while Claude-3.5-Sonnet reaches 94.99% on AITW and 96.21% on AITZ. However, their lower SSR indicates difficulty in mapping these plans to executable screen targets. GPT-5, for instance, records SSR values of only 46.33%, 25.97%, and 19.53% on AITW, AITZ, and CAGUI. By contrast, GUI-oriented models such as Doubao-1.5-UI-TARS and Qwen-2.5-VL-7B achieve much higher SSR, with Qwen-2.5-VL-7B reaching 90.33%, 89.65%, and 83.20% on AITW, AITZ, and CAGUI. Nevertheless, even these stronger grounding-oriented models still fail to complete the full long-horizon task sequences on MaDS-Benchmark.

This contrast highlights a long-horizon scaling bottleneck. The average task length of MaDS-Benchmark is 15.5 steps, which is substantially longer than those of existing public benchmarks. Under such long trajectories, even a single uncorrected grounding deviation may accumulate into terminal failure as the trajectory extends. Therefore, MaDS-Benchmark should be interpreted as a harder stress-test for long-horizon reliability rather than as a benchmark specifically favoring MaDS.

5.2.2 Agent Frameworks

We further compare MaDS with specialized agent frameworks under a unified experimental setting. To ensure a fair comparison and isolate the contribution of system architecture, all frameworks use Doubao-1.5-UI-TARS (Seed, 2025) as the visual encoder and GPT-4o (OpenAI, 2024) for reasoning. In addition, all frameworks are provided with an identical set of successful historical trajectories. Under these conditions, MaDS achieves the strongest overall performance across the evaluated benchmarks.

On MaDS-Benchmark, MaDS reaches a TSR of 90.23%, which is substantially higher than AutoGLM at 71.43% and AgentCPM-GUI at 42.86%. MaDS also maintains an SPSR of 95.83%, indicating stronger long-horizon planning consistency under dynamic and cross-page workflows. We attribute this gain to the combination of scenario-aligned memory retrieval and pre-execution debate-based verification. The former helps the agent remain aligned with the current task context, while the latter intercepts risky actions before execution and reduces the chance that local grounding deviations develop into terminal failures.

On public benchmarks, MaDS also shows consistently strong results. On AITW, MaDS achieves a TSR of 94.74%, which is 13.66% higher than the

strongest compared framework. On AITZ, MaDS reaches 95.24%, exceeding the strongest baseline by 23.81%. On CAGUI, MaDS achieves 96.77%, improving over the strongest compared framework by 32.25%. In addition, on GUIOdyssey, MaDS achieves 92.86% TSR, 96.60% SPSR, and 94.22% SSR, providing further external evidence beyond the proposed benchmark.

Overall, these results show that the advantage of MaDS is not limited to MaDS-Benchmark. MaDS consistently achieves strong performance on AITW, AITZ, CAGUI, and GUIOdyssey, which indicates that the combination of scenario-aligned memory retrieval and pre-execution verification remains effective across multiple GUI agent evaluation settings.

5.2.3 Efficiency Analysis

MaDS improves reliability at the cost of additional inference overhead. Based on our execution logs, MaDS requires 32.66 seconds per step on average and 4083.66 tokens per step. Compared with a single-pass execution setting, the Multi-Round Debate introduces an average marginal overhead of 8.93 seconds per step, while memory retrieval contributes 3.36 seconds per step on average. In addition, the average number of debate rounds is 1.54, indicating that the verification process remains bounded in practice rather than repeatedly cycling for most steps. These results show that the overhead introduced by Multi-Round Debate is measurable but remains controlled in practice (Liu et al., 2026). For long-horizon workflows with irreversible operations, we view this additional cost as a necessary trade-off for improved reliability and early error interception.

5.3 Ablation Study

To analyze the contribution of each component, we conducted an ablation study on a representative subset of MaDS-Benchmark (Table 4).

Operational Experience Memory: Comparing the Baseline (Row 1) with the configuration using only Operational Experience Memory (Row 3), the TSR increases from 5.56% to 72.22%. This result shows that task-specific experience substantially improves long-horizon consistency. By retrieving prior demonstrations and avoiding previously recorded warnings, the agent better maintains the correct trajectory, which also raises the SPSR to 82.94%.

Semantic Knowledge Memory: Adding Semantic Knowledge Memory (Row 5) further raises the TSR to 91.67%, while using this module alone (Row 2) results in only 19.44% TSR. This result indicates that semantic priors alone are insufficient for full task completion, but they effectively complement operational experience by supporting reasoning on unfamiliar interface components and providing general interaction constraints for verification.

Multi-Round Debate: Removing Multi-Round Debate (Row 4 vs. Row 5) leads to a substantial performance drop. The configuration without debate performs worse than the setting using only Operational Experience Memory, which suggests that memory alone is insufficient when risky actions are not explicitly verified before execution. The large RSR gap of 36.20% versus 85.71% further shows that debate plays a critical role in pre-execution verification, helping reject risky actions before local errors propagate into irreversible failures.

6 Conclusion

In this paper, we presented MaDS, a collaborative multi-agent framework designed to improve the reliability of GUI agents in dynamic and long-horizon task automation. By synergizing a **Dual-Layer Memory Module**, which decouples broadly applicable interaction priors from scenario-specific operational experience, with **Multi-Round Debate** for pre-execution verification, MaDS effectively addresses the critical challenges of visual grounding precision and irreversible error accumulation. Extensive evaluations on MaDS-Benchmark, public GUI benchmarks, and GUIOdyssey demonstrate that our system achieves strong performance, validating its effectiveness on long-horizon and dynamic mobile GUI workflows.

Future work will extend the framework to the domain of automated GUI testing. We aim to leverage the system's adversarial evaluation capability, which is currently used for self-verification, to actively probe for UI rendering errors and functional regressions. This direction seeks to evolve Multi-Round Debate from a verifier of decision quality into a more general testing mechanism for identifying edge-case bugs in application development workflows.

Limitations

While MaDS demonstrates strong performance on MaDS-Benchmark and public benchmarks, several

limitations and directions for further improvement remain.

First, the current architectural focus on verification prioritizes safety and success rates, leaving room for efficiency improvements in lower-stakes scenarios. Although Multi-Round Debate effectively intercepts hallucinations in complex and irreversible workflows, the system's inference process is not yet fully optimized for high-throughput or latency-sensitive settings where lightweight responses may suffice. Future work may explore adaptive efficiency mechanisms that dynamically modulate verification depth based on task complexity and execution risk.

Second, the performance ceiling of MaDS remains fundamentally bounded by the perceptual and reasoning limits of the underlying foundation models. While Reflection improves long-term adaptability, it cannot fully eliminate the hallucinations of current MLLMs. Residual errors may still persist in highly ambiguous visual contexts or scenarios requiring domain-specific knowledge, where the model may misinterpret validation signals. Without an external oracle, such uncorrected edge-case successes could gradually reduce the precision of the Operational Experience Memory over extended deployment cycles. A more robust memory sanitation strategy will therefore be important for improving long-term stability.

Third, the current study primarily validates MaDS within the Android ecosystem. Android was selected as the primary platform because its open-source environment and mature automation interfaces, such as ADB, provide the technical support required to verify complex causal logic in long-horizon tasks. Although the Semantic Knowledge Memory is architecturally designed as a pluggable and extensible module, and could in principle be adapted by replacing the current priors with platform-specific guidelines such as the iOS Human Interface Guidelines, its empirical performance and potential gains on iOS or desktop environments remain to be verified through future large-scale experiments. Accordingly, the current evidence should be interpreted as demonstrating generalization across the evaluated mobile GUI benchmarks rather than full cross-platform validation.

Ethical Considerations

The deployment of MaDS in real-world environments entails critical considerations regarding User Privacy and Operational Safety. While mechanisms like Abstract Process Signatures (S_{sig}) reduce reliance on raw pixels, the system inevitably processes visual evidence (I_t, I_{t+1}) that may contain Personally Identifiable Information (PII). To address this, we implemented strict protocols: For the construction of MaDS-Benchmark and all experiments, we utilized controlled dummy accounts and synthesized mock data. To ensure strict compliance and data safety, researchers manually reviewed all visual evidence and applied blurring to any regions containing potentially sensitive patterns before the data entered the Memory Module or was transmitted to cloud-based MLLMs. This ensures that only verified, anonymized visual data is used for reasoning and storage. Despite the pre-emptive verification provided by the Multi-Round Debate, the probabilistic nature of LLMs means that zero-error execution in irreversible tasks cannot be guaranteed with absolute certainty. Consequently, we recommend retaining a "Human-in-the-loop" mechanism for final confirmation in high-stakes decision-making nodes, ensuring that the agent remains an assistant rather than an autonomous executor in critical scenarios.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62407038) and was carried out at the International Design Institute of Zhejiang University.

References

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. 2025. Agent s: An open agentic framework that uses computers like a human. In *The Thirteenth International Conference on Learning Representations*.

Anonymous. 2025. Memoryfield: Exploiting gravitational field for long-term memory management. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. Screenai: a vision-language model for

ui and infographics understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. 2021. Mobile app tasks with iterative feedback (motif): Addressing task feasibility in interactive visual environments. *Preprint*, arXiv:2104.08560.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*.

Chaoran Chen, Zhiping Zhang, Ibrahim Khalilov, Bingcan Guo, Simret A Gebreegziabher, Yanfang Ye, Ziang Xiao, Yaxing Yao, Tianshi Li, and Toby Jia-Jun Li. 2025. Toward a human-centered evaluation framework for trustworthy llm-powered gui agents. *arXiv preprint arXiv:2504.17934*.

Xuweiyi Chen, Ziqiao Ma, Xuejun Zhang, Sihan Xu, Shengyi Qian, Jianing Yang, David F. Fouhey, and Joyce Chai. 2024. Multi-object hallucination in vision language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Weihua Cheng, Ersheng Ni, Wenlong Wang, Yifei Sun, Junming Liu, Wangyu Shen, Yirong Chen, Botian Shi, and Ding Wang. 2025. Mga: Memory-driven gui agent for observation-centric interaction. *Preprint*, arXiv:2510.24168.

Google DeepMind. 2025. Gemini 3 pro best for complex tasks and bringing creative concepts to life. https://deepmind.google/models/gemini/pro.

Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, Rui Yan, and Shuo Shang. 2024. Mobile-bench: An evaluation benchmark for LLM-based mobile agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8813–8831, Bangkok, Thailand. Association for Computational Linguistics.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: towards a generalist agent for the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Android Developers. 2024. Material design for android. https://developer.android.com/develop/ui/views/theming/look-and-feel.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *The Thirteenth International Conference on Learning Representations*.

Ziyi Guan, Jason Chun Lok Li, Zhijian Hou, Pingping Zhang, Donglai Xu, Yuzhi Zhao, Mengyang Wu, Jinpeng Chen, Thanh-Toan Nguyen, Pengfei Xian, Wenao Ma, Shengchao Qin, Graziano Chesi, and Ngai Wong. 2025. KG-RAG: Enhancing GUI agent decision-making via knowledge graph-driven retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5396–5405, Suzhou, China. Association for Computational Linguistics.

Izzeddin Gur, Hiroki Furuta, Austin V Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A real-world webagent with planning, long context understanding, and program synthesis. In *The Twelfth International Conference on Learning Representations*.

Chuanyang Hong and Qingyun He. 2025. Enhancing memory retrieval in generative agents through llm-trained cross attention networks. *Frontiers in Psychology*, 16:1591618.

Jakub Hoscilowicz and Artur Janicki. 2025. ClickAgent: Enhancing UI location capabilities of autonomous agents. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 471–476, Avignon, France. Association for Computational Linguistics.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. Memory in the age of ai agents. *Preprint*, arXiv:2512.13564.

Hongrui Jia, Chaoya Jiang, Shikun Zhang, and Wei Ye. 2025. Decoupling reasoning and perception: An llm-lmm framework for faithful visual reasoning. *Preprint*, arXiv:2509.23322.

Taewoon Kim, Michael Cochez, Vincent Francois-Lavet, Mark Neerincx, and Piek Vossen. 2023. A machine with short-term, episodic, and semantic memory systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(1):48–56.

Quyu Kong, Xu Zhang, Zhenyu Yang, Nolan Gao, Chen Liu, Panrong Tong, Chenglin Cai, Hanzhang Zhou, Jianan Zhang, Liangyu Chen, Zhidan Liu, Steven Hoi, and Yue Wang. 2025. Mobileworld: Benchmarking autonomous mobile agents in agent-user interactive and mcp-augmented environments. *Preprint*, arXiv:2512.19432.

Guangyi Liu, Pengxiang Zhao, Yaozhen Liang, Qinyi Luo, Shunye Tang, Yuxiang Chai, Weifeng Lin, Han Xiao, WenHao Wang, Siheng Chen, and 1 others. 2026. Memgui-bench: Benchmarking memory of mobile gui agents in dynamic environments. *arXiv preprint arXiv:2602.06075*.

Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, Junjie Gao, Junjun Shan, Kangning Liu, Shudan Zhang, Shuntian Yao, Siyi Cheng, Wentao Yao, Wenyi Zhao, Xinghan Liu, and 11 others. 2024. Autoglm: Autonomous foundation agents for guis. *Preprint*, arXiv:2411.00820.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. 2025. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22404–22414.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024. Omniparser for pure vision based gui agent. *Preprint*, arXiv:2408.00203.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2024. CoCo-agent: A comprehensive cognitive MLLM agent for smartphone GUI automation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9097–9110, Bangkok, Thailand. Association for Computational Linguistics.

Siddharth Nayak, Adelmo Morrison Orozco, Marina Ten Have, Vittal Thirumalai, Jackson Zhang, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, Brian Ichter, Anuj Mahajan, and Hamsa Balakrishnan. 2024. Long-horizon planning for multi-agent robots in partially observable environments. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, and 11 others. 2025. GUI agents: A survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22522–22538, Vienna, Austria. Association for Computational Linguistics.

Feiyue Ni, Yanchu Guan, Yuchong Sun, Dong Wang, Chenyi Zhuang, Jinjie Gu, and Ruihua Song. 2025. Rega: Reasoning andnbsp;grounding decoupled gui navigation agents. In *Natural Language Processing and Chinese Computing: 14th National CCF Conference, NLPCC 2025, Urumqi, China, August 7–9, 2025, Proceedings, Part I*, page 375–387, Berlin, Heidelberg. Springer-Verlag.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o.

OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5.

Joonhyung Park, Peng Tang, Sagnik Das, Srikar Appalaraju, Kunwar Yashraj Singh, R. Manmatha, and Shabnam Ghadar. 2025. R-VLM: Region-aware vision language model for precise GUI grounding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9669–9685, Vienna, Austria. Association for Computational Linguistics.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *Preprint*, arXiv:2408.07199.

Qwen. 2025. Qwen2.5 vl! qwen2.5 vl! qwen2.5 vl! https://qwen.ai/blog?id=qwen2.5-vl.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2025. Androidworld: A dynamic benchmarking environment for autonomous agents. *Preprint*, arXiv:2405.14573.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Android in the wild: a large-scale dataset for android device control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Jaechul Roh, Varun Gandhi, Shivani Anilkumar, and Arin Garg. 2025. Break-the-chain: Reasoning failures in llms via adversarial prompting in code generation. *arXiv preprint arXiv:2506.06971*.

ByteDance Seed. 2025. Ui-tars-1.5. https://seed-tars.com/1.5.

Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Enting Chen, Damien Graux, Andre Melo, Ruofei Lai, Zeren Jiang, Zhongyang Li, Ye Qi, Yang Ren, Dandan Tu, and Jeff Z. Pan. 2025. GeAR: Graph-enhanced agent for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12049–12072, Vienna, Austria. Association for Computational Linguistics.

Fei Tang, Haolei Xu, Hang Zhang, Siqi Chen, Xingyu Wu, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Zeqi Tan, Yuchen Yan, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. A survey on (m)llm-based gui agents. *Preprint*, arXiv:2504.13865.

Xingjian Tao, Yiwei Wang, Yujun Cai, Zhicheng Yang, and Jing Tang. 2025. Understanding gui agent localization biases through logit sharpness. *arXiv preprint arXiv:2506.15425*.

Shizuo Tian, Hao Wen, Yuxuan Chen, Jiacheng Liu, Shanhui Zhao, Guohong Liu, Ju Ren, Yunxin Liu, and Yuanchun Li. 2025. Agentprog: Empowering long-horizon gui agents with program-guided context management. *Preprint*, arXiv:2512.10371.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076, Bangkok, Thailand. Association for Computational Linguistics.

Sagar Gubbi Venkatesh, Partha Talukdar, and Srini Narayanan. 2023. Ugif: Ui grounded instruction following. *Preprint*, arXiv:2211.07615.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024b. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15840–15853, Bangkok, Thailand. Association for Computational Linguistics.

Yiqin Wang, Haoji Zhang, Jingqi Tian, and Yansong Tang. 2025a. Ponder & press: Advancing visual GUI agent towards general computer control. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1461–1473, Vienna, Austria. Association for Computational Linguistics.

Ziwei Wang, Leyang Yang, Xiaoxuan Tang, Sheng Zhou, Dajun Chen, Wei Jiang, and Yong Li. 2025b. History-aware reasoning for gui agents. *Preprint*, arXiv:2511.09127.

Yuyang Wanyan, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Jiabo Ye, Yutong Kou, Ming Yan, Fei Huang, Xiaoshan Yang, and 1 others. 2025. Look before you leap: A gui-critic-r1 model for pre-operative error diagnosis in gui automation. *arXiv preprint arXiv:2506.04614*.

Zishu Wei, Qixiang Ma, Xavier Hu, Yuhang Liu, Hui Zang, Yudong Zhao, Tao Wang, Shengyu Zhang, and Fei Wu. 2025. Gair: Gui automation via information-joint reasoning and group reflection.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 543–557.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025a. DiMo-GUI: Advancing test-time scaling in GUI grounding via modality-aware visual reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26257–26267, Suzhou, China. Association for Computational Linguistics.

Qinzhuo Wu, Pengzhi Gao, Wei Liu, and Jian Luan. 2025b. BacktrackAgent: Enhancing GUI agent with error detection and backtracking mechanism. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4250–4272, Suzhou, China. Association for Computational Linguistics.

Qinzhuo Wu, Weikai Xu, Wei Liu, Tao Tan, Liujian Liujianfeng, Ang Li, Jian Luan, Bin Wang, and Shuo Shang. 2024. MobileVLM: A vision-language model for better intra- and inter-UI understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10231–10251, Miami, Florida, USA. Association for Computational Linguistics.

Wenyi Wu, Kun Zhou, Ruoxin Yuan, Vivian Yu, Stephen Wang, Zhiting Hu, and Biwei Huang. 2025c. Auto-scaling continuous memory for gui agent. *Preprint*, arXiv:2510.09038.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025d. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *Preprint*, arXiv:2504.15965.

Mingzhe Xing, Rongkai Zhang, Hui Xue, Qi Chen, Fan Yang, and Zhen Xiao. 2024. Understanding the weakness of large language model agents within a complex android environment. *Preprint*, arXiv:2402.06596.

Zhao Xinjie, Fan Gao, Xingyu Song, Yingjian Chen, Rui Yang, Yanran Fu, Yuyang Wang, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. ReAgent: Reversible multi-agent reasoning for knowledge-enhanced multi-hop QA. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4089, Suzhou, China. Association for Computational Linguistics.

Nan Xu, Fei Wang, Ben Zhou, Bangzheng Li, Chaowei Xiao, and Muhao Chen. 2024. Cognitive overload: Jailbreaking large language models with overloaded logical thinking. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3526–3548, Mexico City, Mexico. Association for Computational Linguistics.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025a. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025b. Aguvis: Unified pure vision agents for autonomous GUI interaction. In *Forty-second International Conference on Machine Learning*.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Silvio Savarese, Caiming Xiong, and Junnan Li. 2025a. Gta1: Gui test-time scaling agent. *Preprint*, arXiv:2507.05791.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025b. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22418–22433, Vienna, Austria. Association for Computational Linguistics.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, and 1 others. 2025. Mobile-agent-v3: Fundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*.

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, and 1 others. 2024a. Large language model-brained gui agents: A survey. *arXiv preprint arXiv:2411.18279*.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025a. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Jiwen Zhang, Jihao Wu, Yihua Teng, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. 2024b. Android in the zoo: Chain-of-action-thought for gui agents. *Preprint*, arXiv:2403.02713.

Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li, and Mengwei Xu. 2024c. Llamatouch: A faithful and scalable testbed for mobile ui task automation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA. Association for Computing Machinery.

Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P Bigham. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024d. Ask-before-plan: Proactive language agents for real-world planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10836–10863, Miami, Florida, USA. Association for Computational Linguistics.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024e. A survey on the memory mechanism of large language model based agents. *Preprint*, arXiv:2404.13501.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, Jie Xie, Wei Zhou, Wang Xu, Yuanheng Zhang, Zhou Su, Zhongwu Zhai, Xiaoming Liu, Yudong Mei, Jianming Xu, and 6 others. 2025b. AgentCPM-GUI: Building mobile-use agents with reinforcement fine-tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 155–180, Suzhou, China. Association for Computational Linguistics.

Di Zhao, Longhui Ma, Siwei Wang, Miao Wang, and Zhao Lv. 2025. Cola: A scalable multi-agent framework for windows ui task automation. *arXiv preprint arXiv:2503.09263*.

Zichen Zhu, Hao Tang, Yansi Li, Dingye Liu, Hongshen Xu, Kunyao Lan, Danyang Zhang, Yixuan Jiang, Hao Zhou, Chenrun Wang, Situo Zhang, Liangtai Sun, Yixiao Wang, Yuheng Sun, Lu Chen, and Kai Yu. 2025. MobA: Multifaceted memory-enhanced adaptive planning for efficient mobile task automation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 535–549, Albuquerque, New Mexico. Association for Computational Linguistics.

A Parameter Specifications

To ensure experimental reproducibility, we detail the hyperparameter settings for the components of the MaDS system in this section. All parameters were calibrated based on a held-out validation set comprising 20 representative long-horizon tasks and remained fixed across all evaluation benchmarks.

A.1 Confidence Scoring in Multi-Round Debate

In the Multi-Round Debate (Section 3.2.1), the final confidence score is calculated as $C_{final} = w_1 C_{rule} + w_2 C_{ground} + w_3 C_{logic}$. To enforce a safety-first principle during GUI operations, we adopted a hard-constraint dominant configuration.

The specific parameter settings are presented in Table 5.

A.2 Memory Retrieval and Fallback Mechanism

For the Retrieval within the Memory Module (Section 3.1.2), we implemented a dynamic fallback to

Parameter	Value	Description & Rationale
Weights		
w_1 (C_{rule})	**0.4**	**Weight for Rule Compliance.** We assign a high weight to ensure that any violation of universal interaction norms (e.g., performing a 'type' action on a non-input widget) is strictly penalized.
w_2 (C_{ground})	**0.4**	**Weight for Visual Grounding.** We assign a high weight to align with the binary nature of C_{ground} (0 or 1). This ensures that operations targeting hallucinated (non-existent) elements are effectively intercepted.
w_3 (C_{logic})	**0.2**	**Weight for Logical Consistency.** This weight allows the system to distinguish which action is contextually optimal among those that are already compliant and physically valid.
Thresholds		
τ_{conf}	**0.8**	**Consensus Threshold.** Setting this threshold to 0.8 creates a mathematical "**Veto Mechanism**." Since $w_1 + w_3 = 0.6 < 0.8$ and $w_2 + w_3 = 0.6 < 0.8$, if either C_{rule} (Rule Check) or C_{ground} (Visual Check) evaluates to 0, the final score C_{final} cannot exceed τ_{conf} regardless of the logic score. This forces the system to reject the plan and trigger the Rebuttal process.
K_{max}	**3**	**Maximum Debate Rounds.** We limit the debate to a maximum of 3 rounds to prevent excessive computational costs associated with infinite reasoning loops.

Table 5: Parameter specifications for the Multi-Round Debate mechanism.

balance retrieval precision and recall.

Retrieval ($TopK$): Set to **5**. We retrieve the top-5 most relevant experiences. This number was empirically chosen to provide sufficient context for the Planning Agent while preventing information overload and maintaining low inference latency.

Retrieval Similarity Threshold (τ_{recall}): Set to **0.75**. The system first attempts to retrieve operational experiences within the specific scope defined by the current scenario index L_{scene}. If the vector similarity score of the Top-1 retrieved result falls below the threshold ($Sim < 0.75$), the system infers that the current scenario-specific memory is insufficient. Consequently, it automatically disables the scenario constraint and falls back to the global memory pool. This ensures that experiences are not excluded due to potential granularity issues in scenario clustering.

A.3 Memory Maintenance Strategy

Algorithm 1 demonstrates how memory is maintained.

B Composition of Semantic Knowledge Memory

Table 6 shows each of the three dimensions of Semantic Knowledge Memory contains.

C Prompts Templates

We provide the prompts templates used in the MaDS. Note that $\{\{variable\}\}$ denotes dynamic slots filled at runtime.

C.1 Planning Agent Prompt

The Planning Agent utilizes both Semantic Knowledge Memory and retrieved Operational Experience Memory to generate the next step-level instruction.

Core Principles

Atomicity: Each task must describe only one immediate operation. Do not merge multiple steps.
Completeness: Strictly follow the logical order of the global task. Never skip any necessary steps required by the global instruction.
Specificity: Always prioritize specific element names, titles, or content descriptions over positional indices (e.g., "Click the 'Settings' icon" instead of "Click the first icon").

Prompt 1

System Instruction:
You are a UI task decomposition expert.
Global Task: `{{whole_task}}`

Task Generation Instruction:
Based on the following context, reason and generate the [Single-Step Interaction Task] for the current page:

Input Context:
Current Screenshot: (Image Uploaded)
History Summary: `{{history_summary}}`
Relevant Experience: `{{experience_guidance}}`

Output Requirement:
Directly output the final task description string. Do

Algorithm 1 Memory Maintenance Strategy

Require: New experience node v_{new} (from Reflection phase)
Require: Operational Experience Memory $\mathcal{M}_{exp}$
Require: Thresholds τ_{sim}, τ_{prune}, Weights α, β, λ

Stage 1: Hard Deduplication ▷ *Filtering redundant Reflection outputs*

$Scenario \leftarrow v_{new}.L_{scene}$
if $\exists v \in \mathcal{M}_{exp}[Scenario]$ s.t. $v.\mathcal{F}_{out} = v_{new}.\mathcal{F}_{out}$ $\wedge\ v.Label_{suc} = v_{new}.Label_{suc}$ **then**
  return ▷ Discard: Exact match found
end if

Stage 2: Semantic Fusion ▷ *Integrating new insights*

$\mathbf{e}_{new} \leftarrow \text{Embed}(v_{new}.D_{task} \oplus v_{new}.S_{sig})$
$v_{top}, d \leftarrow \text{RetrieveTop1}(\mathbf{e}_{new}, \mathcal{M}_{exp}[Scenario])$
$Sim \leftarrow 1 - d$
if $Sim < \tau_{sim}$ **then**
  $\mathcal{M}_{exp} \leftarrow \mathcal{M}_{exp} \cup \{v_{new}\}$ ▷ Case 1: Novel experience
else
  $v_{merged} \leftarrow \text{LLM\_Fusion}(v_{new}, v_{top})$
  $\mathcal{M}_{exp} \leftarrow (\mathcal{M}_{exp} \setminus \{v_{top}\}) \cup \{v_{merged}\}$ ▷ Case 2: Logic fusion or optimization
end if

Utility-Based Pruning ▷ *Removing stale data*

for all $v \in \mathcal{M}_{exp}$ **do**
  $U_t(v) \leftarrow \alpha \cdot N_{access}(v) + \beta \cdot e^{-\lambda \Delta t}$
  if $U_t(v) < \tau_{prune} \wedge v.Label_{suc} \neq \text{Fail}$ **then**
    $\mathcal{M}_{exp} \leftarrow \mathcal{M}_{exp} \setminus \{v\}$ ▷ Prune low-utility nodes (Strictly preserve Warnings)
  end if
end for

not include extra explanations.

Special Scenario Constraints:
1. [Specific Element Recognition Requirements]
- When the task involves clicking an element at a specific position (e.g., "Click the second product", "Select the third option"), you must carefully observe the specific content in the screenshot.
- The subtask description must contain specific element information rather than ordinal descriptions.
- *Incorrect:* "Click the second product card"
- *Correct:* "Click the 'OPPO Reno5' product card"

2. [Component Visibility & Scrolling Planning]
- When clicking a component, first determine if it is "fully visible" in the current screenshot (not obstructed and not outside screen boundaries).
- If the component is not fully displayed, partially obstructed, or outside the viewport, the current subtask must be planned as a scroll operation (Scroll Up/Down) to bring the component into full view.

3. [Pop-up/Overlay Handling Rules]
- If the interface displays a pop-up, permission request, advertisement overlay, update prompt, login/registration window, or guidance mask that is irrelevant to the global task and blocks execution, the current step must be prioritized to Close/Reject/Dismiss that pop-up.

C.2 Evaluation Agent - Multi-Round Debate Prompt

This prompt corresponds to the Multi-Round Debate described in Section 3.2.1. Before execution, the Evaluation Agent scrutinizes the plan based on constraints and logical consistency.

Core Principles

Evaluate the next step instruction generated in the previous step using three criteria listed in 3.2.1.

Prompt 2

Input Context:
Current Screenshot: (Image Uploaded)
History Summary: `{{history_summary}}`
Relevant Experience: `{{experience_guidance}}`
Step-level instruction generated by Planning Agent: `{{instruction}}`

Table 6: Three Dimensions of Semantic Knowledge Memory.

Category	Definition & Mechanism	Typical Examples / Parameters
I. Physical Boundaries of Action Space $\mathcal{A}$	**Mechanism:** Defines the legal bounds of interaction as hard constraints. We utilize parameterized functions to anchor outputs to real UI elements. **Rationale:** Strictly prevents *action hallucinations* common in open-ended generation by enforcing a grounding verification step.	• `click(box=[x,y,w,h])` • `double_tap(box=[x,y,w,h])` • `long_press(box=[x,y,w,h])` • `scroll(start_box, end_box, direction)` • `drag(start_box, end_box)` • `type(content="text")` • `hotkey(key="name")` • `wait / finished`
II. Visual-Semantic Mappings	**Mechanism:** Establishes mappings between visual features and functional semantics. **Crucial Design:** We treat these mappings as **Probabilistic Priors** rather than rigid rules. **Contextual Overriding:** Real-world designs may deviate. During the *Multi-Round Debate*, if strong contextual evidence contradicts the prior, the specific visual context takes precedence.	• `Color(Red/Orange)` → Warning • `Icon(Gear)` → Settings • `Icon(Magnifier)` → Search • `Animation(Spinner)` → Loading • ...
III. General Interface Transition Rules	**Mechanism:** Encapsulates the logic of interface flow and standard sequential procedures. **Rationale:** Endows the system with *cold-start reasoning capabilities*, allowing it to handle routine tasks (e.g., login, pop-ups) via logical inference even in the absence of specific operational experience.	• **Login:** Input User → Pass → Click • **Cold-Start:** Launch → Privacy Pop-up → Close • **Navigation:** Sub-page → Back → Home • ...

Output Format:

```
EVAL:
{
   "decision": "PASS" | "REJECT",
   "confidence_score": 0.0-1.0,
    "rationale":    "If  REJECT,  detailed
specific  reasons,  pointing  out  which
criterion was violated, and providing a
correction strategy."
}
TASK:
<The final step-level operation instruction.
Only one line. Do not merge steps.>
```

C.3 Action Agent Prompt

The Action Agent receives the verified instruction and converts natural language into precise physical coordinates or ADB commands using the following prompt.

Prompt 3

Input Context:
Action Space shown in Table 6
Step-level instruction evaluated by Evaluation Agent: `{{instruction}}`

[Direct Input Rules]

- When the subtask is "Input xxx", directly use the command: `type(content='xxx')`.
- Do not click the input box first; execute the input operation directly.

[Visibility Check]

- If the target component is not fully displayed, partially obstructed, or outside the viewport, the current step must correspond to a **Scroll** operation (Up/Down/Left/Right) to reveal the component completely.
- Execute the **Click** operation ONLY when the component is fully visible.
- If overlays/pop-ups exist, clear them according to the Pop-up Handling Rules before scrolling or clicking.

[Scroll/Drag Operation Output Format]

- You must output both the start and end coordinates:
  `start_box=[x1, y1, x2, y2], end_box=[x3, y3, x4, y4]`

C.4 Evaluation Agent - Reflection Prompt

After action execution, the Evaluation Agent diagnoses whether the current step-level operation was successfully executed by comparing screenshots before and after the operation.

Prompt 4

System Instruction:
You are a UI automation diagnosis expert. Your task is to analyze the execution result of the previous step.

Input Context:
Below is the detailed information of this operation:

```
[Global Task]: {global_task}
[Current    Step-level    Instruction]:
{current_step-level_instruction}
```

[Executed Action]: {action_type}
[Reasoning Thought]: {thought}
Pre-action Screenshot: {image_before}
Post-action Screenshot: {image_after}

Output Format Constraint:
Please output ONLY a pure JSON object. Do not include Markdown code blocks or extra text:
{'success': True | False, 'reason': 'Detailed explanation of the judgment.'}

D Benchmarks

D.1 Comparison of Public Benchmarks

Table 7 details the key statistics, task categories, evaluation purposes, and collection methodologies of various datasets. As shown in the table, most existing benchmarks primarily focus on outcome evaluation, measuring the success rate of task completion. In contrast, our proposed **MaDS-Benchmark** is designed for *process-oriented evaluation*, analyzing the underlying causes of agent failures. Furthermore, our collection method leverages a approach of human-defined tasks, automated collection and annotation, then human check.

D.2 MaDS Diagnostic Evaluation

The data captured in structured log (*history.jsonl*) are as followed:

Visual Context (I_t, I_{t+1}): The screenshot pair capturing the interface state immediately before and after the action, superimposed with grounding markers.

Perception: Logs the semantic abstraction of the screen I_t, including the parsed UI elements and the identified L_{scene}.

Retrieval Context: Archives the Dual-Track Retrieval results, including the universal priors from Semantic Knowledge Memory and the retrieved memory from Operational Experience Memory.

Planning & Debate: Records the finalized plan P'_t along with the full Multi-Round Debate history, including the Rebuttal Rationale provided by the Evaluation Agent and the C_{final}.

Action Execution (A_t): Documents the concrete ADB commands executed.

Reflection: Records the $Label_{suc}$ and $Reason$ generated by the Evaluation Agent, encapsulating the system's self-assessment of the outcome.

D.3 Selection of Benchmarks for Experiments

To ensure a comprehensive evaluation of the MaDS system, we selected three public benchmarks from Table 7 that represent different challenges in GUI automation for our experiment: scale, application diversity, and layout complexity. Given the diverse task categories encompassed within these benchmarks, we specifically sampled tasks focused on Android mobile applications for our experiments.

D.3.1 Android In The Wild (AITW)

Source & Scale: Proposed by Google Research, AITW (Rawles et al., 2023) is currently the largest dataset for GUI agents. It comprises 715,000 episodes spanning 30,000 unique natural language instructions. The dataset captures interactions across four distinct subsets:

- **GoogleApps:** Interactions within the Google ecosystem (Gmail, Calendar, Photos, etc.).
- **Install:** Tasks involving app installation and login permissions.
- **WebShopping:** E-commerce tasks on browser-based shopping websites.
- **General:** Miscellaneous tasks across various third-party applications.

Reason for Selection: We selected AITW to evaluate the system's fundamental capability to handle standard, English-language tasks. Its massive scale and diverse instructional phrasing allow us to benchmark MaDS against SOTA models in a standardized environment.

D.3.2 Android In The Zoo (AITZ)

Source & Scale: AITZ (Zhang et al., 2024b), introduced by Zhang et al., focuses on essential tasks in the Android ecosystem. It contains 2,504 interaction episodes comprising 18,643 screen-action pairs. Distinct from AITW's focus on Google apps, AITZ explicitly targets a diverse set of 71 third-party applications (e.g., Spotify, WhatsApp, Uber) covering widely-used categories such as Communication, Media, and Navigation. The dataset features 478 unique task types, collected across multiple device sizes to introduce resolution variance.

Reason for Selection: We incorporated AITZ to evaluate generalization in various applications. While AITW is dominated by Google's design language, AITZ exposes the agent to the heterogeneous design patterns of third-party developers. This benchmark is crucial for testing whether the *Semantic Knowledge Memory* in MaDS can effectively transfer universal interaction priors across visually distinct applications that the agent may not have encountered before.

Benchmark	#Apps	#Ep.	Avg. Steps	Task Categories	Purpose	Collection & Annotation	Key Annotations
AITW (Rawles et al., 2023)	159	715k	6.5	GoogleApps, Install, Webshopping, General, Single	Training	Human Collection + Human Annotation	Instructions, Screenshots, Actions, OCR Screen Features
AITZ (Zhang et al., 2024b)	70+	2504	7.5	General, GoogleApps, Install, WebShopping, Single	Training & Evaluation (Zero-shot & Fine-tuning)	Human Collection + GPT-4V Generation with Human Verification and Refinement	Screen Description, Actions (Including Thinking)
AndroidWorld (Rawles et al., 2025)	20	116	/	Productivity, Communication, Multimedia	Evaluation (Outcome)	Programmatic Generation + Verification	System State, UI Elements
AndroidArena (Xing et al., 2024)	13	221	6.13/11.14/6.03	Single-App, Cross-App, Constrained Tasks	Evaluation (Outcome)	Human Collection + Human Annotation	Instructions, Action Sequences, Constraints
LlamaTouch (Zhang et al., 2024c)	57	496	7.01	Tools, Social, Shopping, General, Install, Entertainment	Evaluation (Outcome)	Human Collection + Human Annotation	UI State, System State, Action, View Hierarchy
AppWorld (Trivedi et al., 2024)	11	750	/	Difficulty Levels	Evaluation (Outcome)	Programmatic Gen. + Verification	Instructions, System State, Solutions
CAGUI (Zhang et al., 2025b)	30+	600	7.5	Visual Grounding, Action Prediction	Evaluation (Outcome)	Human Collection + LLM-Assisted	Instructions, Screenshots, Actions, XML
MaDS-Benchmark (Ours)	**11**	**271**	**15.5**	Content, E-commerce, Services (Dynamic Scenarios)	**Evaluation (Diagnostic)**	Human Defined Tasks + Automated Trace Recording	**Diagnostic Logs (Section 4.2)**

Table 7: Comparison of MaDS-Benchmark with existing GUI automation benchmarks.

D.3.3 Chinese Android GUI (CAGUI)

Source & Scale: CAGUI (Zhang et al., 2025b) represents the first large-scale benchmark dedicated to the Chinese mobile app ecosystem. It consists of 55,000 task trajectories with over 470,000 action steps. The dataset covers 30+ mainstream Chinese applications (e.g., WeChat, Douyin, Meituan) across eight domains including Life Services, Finance, and Social Networking. Notably, these applications are characterized by the Super App, featuring significantly denser information layouts, complex nested navigation, and frequent dynamic pop-ups compared to their Western counterparts.

Reason for Selection: We selected CAGUI to serve as a complementary test to our self-constructed MaDS-Benchmark, addressing the need for evaluation:

- **Breadth vs. Depth:** Unlike our MaDS-Benchmark, which specifically targets *long-horizon* (avg. 15+ steps) and highly dynamic live scenarios in selected core scenarios, CAGUI provides extensive coverage breadth. It spans eight distinct domains (including Education and Productivity) that are not fully covered in our specialized benchmark, allowing us to verify the agent's generalization capability across a wider variety of app architectures.
- **Standardized Comparison:** As an established public benchmark, CAGUI allows for a standardized comparison against other agents, mitigating potential bias inherent in self-constructed datasets.
- **Linguistic & Layout Complexity:** It serves as a test for linguistic generalization and layout robustness, verifying that the improvements gained from the *Multi-Round Debate* mechanism are universally applicable and not limited to specific task types.

D.4 Evaluation Metrics

We conduct a fine-grained evaluation across four dimensions:

Final Task Success Rate (TSR): Since the accumulation of successful actions does not guarantee global task completion, we employ human evaluation as the final safeguard. For each long-horizon task, researchers strictly judge whether the task was truly accomplished based on the global task description, the final Post-Action Screenshot, and the actual execution trajectory.

Step-level Planning Success Rate (SPSR): Evaluates the correctness of the decision-making logic (intent and target semantics), serving as a proxy for reasoning capabilities.

Step-level Success Rate (SSR): Evaluates the execution precision (Visual Grounding and Actuation). This metric isolates execution errors from planning errors.

Retry Success Rate (RSR): [Ablation Specific] Defined as the ratio of successful recovery attempts to the total number of retries, quantifying error recovery capability.

E MLLMs and Agent Frameworks Selection

To comprehensively evaluate the performance of MaDS, we structured our comparative experiments into two distinct groups: Foundation MLLMs (End-to-End) and Agent Frameworks (System-Level). This division allows us to isolate the contributions of the architectural design from the raw capabilities of the underlying models.

E.1 Foundation MLLMs

This group aims to verify whether SOTA MLLMs can independently handle long-horizon tasks, thereby demonstrating the necessity of the MaDS architectural design. We selected models representing different capability tiers:

GPT-5 (OpenAI, 2025) / Claude-3.5-Sonnet (Anthropic, 2024): Representing the current pinnacle of *closed-source* model intelligence. We test these to determine if top-tier reasoning capabilities alone are sufficient to overcome the lack of memory and grounding mechanisms.

Qwen-2.5-VL-72b (Qwen, 2025) / Gemini-3-Pro (DeepMind, 2025): Selected for their superior long-context understanding and multimodal processing capabilities, serving as strong baselines for handling the dense information flow in continuous GUI operations.

Doubao-1.5-UI-TARS (Seed, 2025): We utilize the Doubao-1.5-UI-TARS model in a "Screenshot $\rightarrow$ Planning $\rightarrow$ Action" single-step inference mode *without* mounting the MaDS memory module or multi-round debate mechanism. This comparison strictly isolates the performance gain attributed to our proposed architecture.

E.2 Agent Frameworks

We selected three representative Agent frameworks to evaluate MaDS against existing system-level solutions. To eliminate performance variance caused by different visual encoders, we standardized the visual understanding module across all frameworks. We replaced the native visual encoders of AndroidArenaAgent (Xing et al., 2024), AgentCPM-GUI (Zhang et al., 2025b), and AutoGLM (Liu et al., 2024) with the identical **Doubao-1.5-UI-TARS** model used in MaDS.

E.2.1 AndroidArenaAgent (Google DeepMind)

AndroidArenaAgent (Xing et al., 2024) is a pioneering framework that introduced a scalable environment and a unified action space for Android agents. It typically relies on a standard "Observation-Action" loop.

We selected AndroidArenaAgent as a classic baseline for the standard agent paradigm. By upgrading its visual encoder to Doubao-1.5-UI-TARS, we can specifically evaluate whether its linear planning logic falls short compared to MaDS's non-linear debate and memory retrieval mechanisms in long-horizon tasks.

E.2.2 AgentCPM-GUI (OpenBMB)

AgentCPM-GUI (Zhang et al., 2025b) focuses on optimizing smaller models (e.g., 8B parameters) for GUI tasks through Reinforcement Fine-Tuning (RFT) and Chain-of-Thought (CoT) reasoning via a "Critic-Play" mechanism.

This framework represents the "Fine-tuned Specialist" approach. Comparing MaDS against AgentCPM-GUI allows us to verify whether our *training-free* architectural enhancements can outperform a model that has been computationally *fine-tuned* specifically for GUI interactions, given the same visual capabilities.

E.2.3 AutoGLM (Zhipu AI)

AutoGLM (Liu et al., 2024) represents the current SOTA in autonomous GUI agents. It employs a "Type-Agent" and "Touch-Agent" architecture designed for curriculum learning and self-evolution.

As one of the strongest existing autonomous agents, particularly known for its high success rate in cross-application Web and Android environments, AutoGLM serves as the primary competitor. Outperforming it demonstrates MaDS's superiority in handling the specific challenges of complex mobile application scenarios.

F Qualitative Case Studies

F.1 Efficacy of Multi-Round Debate

F.1.1 Case 1 - Clicking a Product Card with a Specific Label

Instruction: Click the card with the "Live" tag to enter the broadcast room.

MaDS, due to its Multi-Round Debate, directly rejected the operation that could not find a valid Visual Grounding, shown in Figure 4.

F.1.2 Case 2 - Inputting Text

Instruction: Input text "iphone" in the search box.

MaDS, due to its Multi-Round Debate, directly rejected operations that did not follow general rules, shown in Figure 5.

F.2 Robustness in Real-World Scenarios

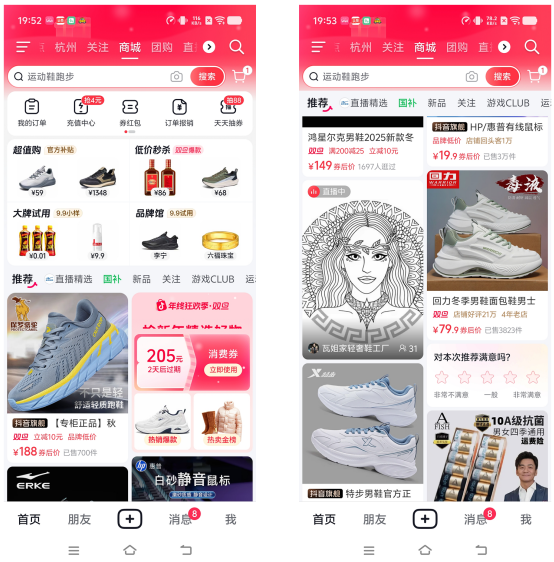
F.2.1 Case 3 - Joining the Fans Club

Instruction: Join the Fans Club.

Dynamic Event: A system prompt appears indicating "Insufficient Diamonds".

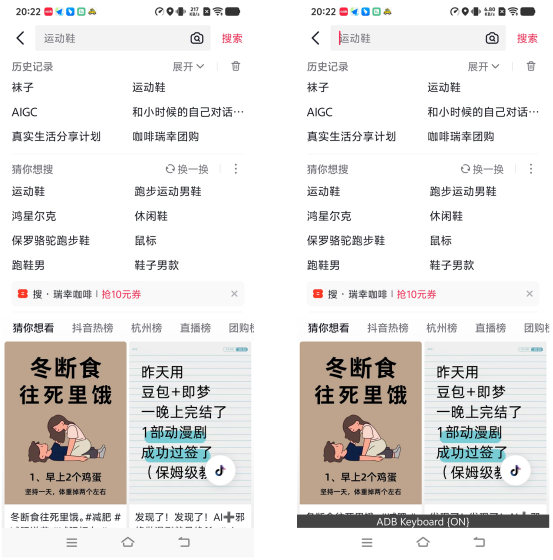
Common Failure: Other models and agents enters a loop of repeatedly clicking "Join" or accidentally exits the interface, failing to recognize that the task is unexecutable.

MaDS identifies operations that cannot be executed due to system reasons and rolls back instead of repeating, shown in Figure 6.



- **Planning Agent (Round 1):** Proposes clicking the target coordinates directly.
- **Evaluation Agent:** REJECT.
  - **Rationale:** Visual Grounding Violation (C_{ground}). The target element is currently not visible.
  - **Planning Agent (Round 2):** Revises the subtask to scroll(direction='up') to reveal the target.

Figure 4: Case 1 Demo



- **Planning Agent (Round 1):** Proposes executing type("iphone").
- **Evaluation Agent:** REJECT.
  - **Rationale:** Violates the universal rule requiring a 'Click-then-Input' sequence for this specific component.
  - **Planning Agent (Round 2):** Revises the task to click(search_box).

Figure 5: Case 2 Demo

F.2.2 Case 4 - Clicking the Tab

Instruction: Click the "Nearby Food" sub-tab.

Dynamic Event: A "New User Coupon" pop-up appears immediately after the page transition.

Common Failure: Other models and agents attempt to click the coordinates of the tab behind the pop-up (occlusion error) or waits indefinitely.

MaDS identifies dynamic pop-ups and prioritizes processing them, shown in Figure 7.

F.3 Impact of Memory Retrieval Mechanism

F.3.1 Case 5 - Sliding the navigation bar and clicking the tab.

Instruction: Click the "Topic" tab.

Without Memory (Zero-shot) The agent fails to locate the "Topic" label because it is hidden off-screen in a scrollable navigation bar. It attempts random clicks or fails to output an ADB command.

MaDS searches the memory and uses operational experience as guidance to find possible operating instructions, shown in Figure 8.

G Experiment Evaluation Artifacts and Protocol

To ensure the rigor and reproducibility of our *Diagnostic Evaluation Protocol*, we adopt a post-hoc auditing approach rather than real-time observation. This allows researchers to scrutinize the decision-making process at a granular level. The evaluation relies on two primary categories of structured artifacts generated by the system.

G.1 Global Task Metadata

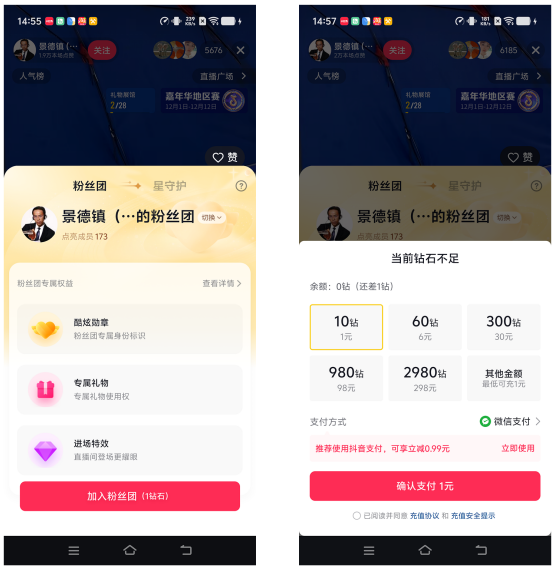
For every evaluation episode, the system generates a `task_info.json` file, shown in Figure 9. This file serves as the Ground Truth for the researchers to evaluate.

The researchers first reads the `global_task` field to understand the full scope of the requirement. Then, we review the complete sequence of screenshots stored in the `img/` directory to determine if the final visual state matches the task completion criteria.

G.2 Step-Level Execution Logs

To diagnose specific failures, differentiating between a logic error and a grounding error. The system archives a detailed trace for every ADB operation.

When visual evidence is ambiguous, evaluators inspect the `action`, `reason`, and `adb_commands` fields. This white-box access allows evaluators to confirm whether the system's internal `thought` process was logical, even if the physical execution failed due to system latency or occlusion. Detailed logs and corresponding annotated screenshots are shown in Figure 10.



- **Perception:** Doubao-1.5-UI-TARS detects the newly appeared pop-up.
- **Planning:** Recognizes the constraint (insufficient balance) prevents joining, and plans to close the prompt to proceed with alternative tasks.
- **Action:** Closes the pop-up window.
- **Outcome:** After closing the pop-up, MaDS plans the next logical task (e.g., "Return to broadcast room").

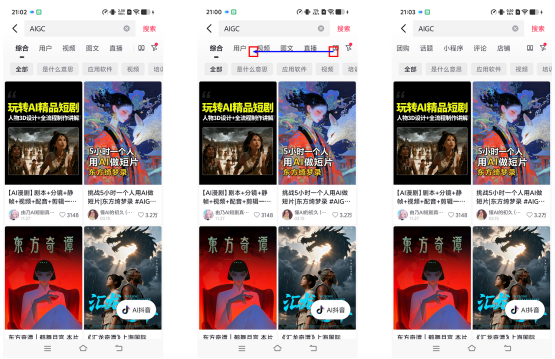
Figure 6: Case 3 Demo

- **Perception:** Doubao-1.5-UI-TARS detects the obstruction caused by the coupon pop-up.
- **Planning:** Closes the pop-up window.
- **Action:** Closes the pop-up window.
- **Outcome:** Once the path is clear, MaDS executes the original instruction "Click 'Nearby Food' sub-tab".

Figure 7: Case 4 Demo

H Human Subjects and Annotation Details

We recruited human annotators to perform two key tasks: (1) **Outcome Verification**, where participants reviewed execution logs to verify Task Success Rate (TSR) and identify failure causes with automated annotations generated by our system; and (2) **Operational Supervision**, where participants monitored the device screens in real-time during the agent's execution to intervene in case of unexpected behaviors.



- **Retrieval:** Based on the scenario label L_{scene}, the system retrieves a high-scoring S_{sig} from Operational Experience Memory.
- **Guidance:** Experience Hint - Swipe left on the top navigation bar to find the topic tag.
- **Result:** The agent executes the swipe operation, reveals the specific tag, and successfully clicks it.

Figure 8: Case 5 Demo

task_info.json

```
{
  "task_id": "csv_task_1_20251201_205700",
  "global_task": "Click Douyin Login -> Click top [Search] icon -> Input 'AIGC' -> Click [Search] -> Swipe left on tabs until [Topic] is visible -> Click [Topic] tab -> Click the first topic -> Swipe up to browse videos -> Click [Participate] button -> Enter shooting page -> Click bottom-left [Album] -> Select an image -> Click [Next] -> Click top-left [Back] -> Click [Give Up] -> Click [Confirm] -> Return to Topic Page",
  "app_name": "unknown_app",
  "created_at": "2025-12-01T20:57:00.829520",
  "status": "in_progress"
}
```

Figure 9: Structure of *task_info.json* (Global Task Metadata)

H.1 Recruitment and Payment

We recruited **5** participants through a public call distributed via internal university blog. All participants were required to have a background in Computer Science or related fields and proficiency in using Android smartphones. We compensated participants at a rate of 60 CNY per hour. The compensation provided is approximately 2× the local minimum hourly wage, which we consider adequate and ethical given the complexity of the tasks.

H.2 Data Consent and Usage

Prior to the start of the tasks, informed consent was obtained from all participants. We provided a digital consent form explaining that:

1. Their annotations (judgments of success/failure) would be used to evaluate the system.
2. The aggregated data would be published in an academic paper and open-source dataset.
3. No personally identifiable information (PII) regarding the annotators themselves would be released.
4. They had the right to withdraw from the study at any time without penalty.

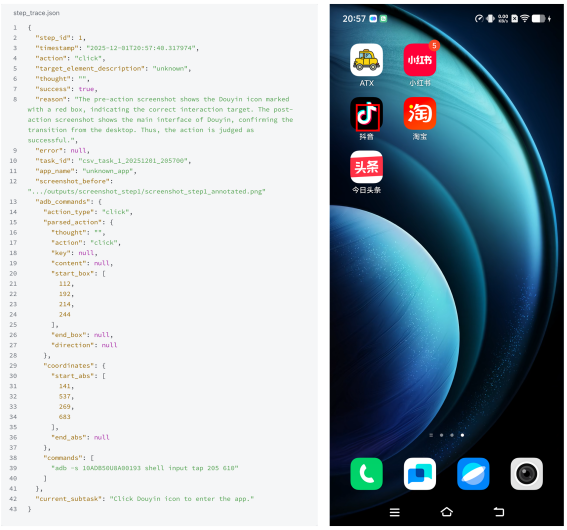


Figure 10: Structure of Step-Level Execution Logs (*history.jsonl*) and Corresponding Annotated Screenshots of the Actions.

Participants were required to digitally sign this form before accessing the annotation platform or the experimental devices.

H.3 Ethics Review Board Approval

The data collection and annotation protocol described in this study is currently under review. Although the formal approval process is ongoing, we have strictly adhered to ethical principles regarding data privacy and subject protection throughout the study. Crucially, all data utilized in this research has already undergone de-identification and desensitization processes prior to any collection and analysis, ensuring that no PII or sensitive content is exposed regardless of the pending administrative review.

H.4 Instructions Given to Participants

We prioritized clarity and safety in our instructions. For the operational supervision task, we specifically emphasized the stop protocol to ensure safety during GUI automation. The full text of the instructions provided to the participants is presented in the following Box.

Full Instructions for Human Annotators (Operational Supervision & Evaluation)

Overview: You will be monitoring an GUI Automation System operating a smartphone to complete long-horizon tasks. Your goal is to (1) Monitor the agent for safety, and (2) Evaluate if the task was successful, double check the automated annotations generated by systems.

Part 1: Safety Monitoring (Real-time)

- Watch the mirrored screen stream carefully.
- **STOP PROTOCOL:** Press the red "Emergency Stop" button immediately if the agent attempts to:
  - Navigate to "Settings" to reset the phone or uninstall apps.
  - Send messages containing gibberish or offensive content.
  - Make a payment.
- **Disclaimer:** The accounts used are dummy test accounts. No real money or real user data is at risk, but we must prevent system-level damage.

Part 2: Outcome Verification (Post-hoc)

- Read the "Task Goal" displayed at the top.
- Review the final screenshot and the step history.
- **Success Criteria:** Mark as "Success" ONLY if the final state matches the goal perfectly.
- **Failure Attribution:** If failed, select the primary reason from the dropdown:
  - *Grounding Error:* Clicked the wrong button.
  - *Logic Error:* Performed a step out of order.
  - *System Error:* App crashed or network failed.

Risks: There are no physical risks associated with this task. To minimize eye strain, please take a 5-minute break every hour.

I Statement the Usage of AI

In accordance with the ACL Policy on AI Writing Assistance, we disclose the scope of generative AI utilization in this work: We used Gemini and ChatGPT to assist with paraphrasing, polishing, and refining the language of the manuscript to enhance clarity and readability. We affirm that all scientific claims, novel ideas, experimental designs, and the overall structure of this paper were conceived solely by the human authors. All content refined by AI tools was rigorously verified by the authors for accuracy and completeness. We assume full responsibility for the correctness and originality of the final manuscript.