

DAC-Bench: A Decision-Aware Benchmark for Compositional Mobile GUI Tasks

Yuqing Zhang^{1*}, Honghui Sheng^{1*}, Xueyu Hu¹, Shengyu Zhang^{1,2 †}, Fei Wu^{1,2 †}

¹Zhejiang University, Hangzhou, China

²Shanghai Institute for Advanced Study of Zhejiang University, Shanghai, China

{yuqingz7, sy_zhang, wufei}@zju.edu.cn

Abstract

Mobile GUI agents powered by LMMs can perceive screens and follow instructions, yet existing benchmarks largely target short, linear workflows and step-level accuracy, offering limited insight into long-horizon planning and decision-making under branching structures. We present DAC-Bench, a decision-aware benchmark with compositional tasks comprising 830 episodes and 11,345 action steps across 35 applications on Android and iOS. Tasks are organized into Sequential, Conjunctive, Conditional, and Hierarchical structures, reflecting real-world multi-step and branching interaction patterns. To complement standard step-level evaluation, we introduce weighted longest common subsequence to capture length-sensitive progress and decision accuracy for branch correctness. Evaluations across 7 diverse agents show substantial performance degradation compared to prior benchmarks, with success rates dropping below 5% on 6–8 step tasks and branch accuracy averaging 38%, highlighting challenges in conditional decision-making. By exposing these failure modes, DAC-Bench provides a challenging and diagnostic benchmark for advancing decision-aware mobile GUI agents. Our code and dataset are available at: <https://github.com/YuqingZhangMirror12/DAC-Bench>.

1 Introduction

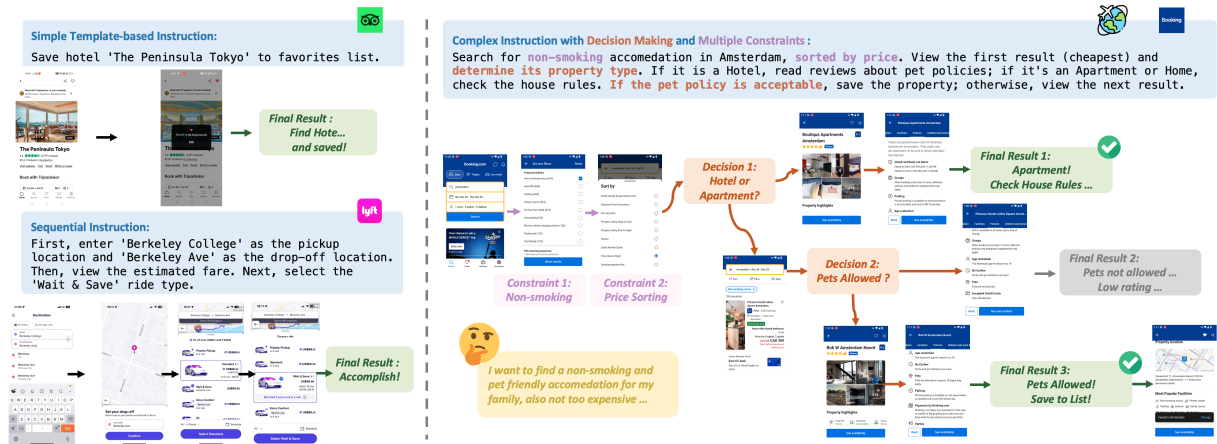
Recent advances in Large Multimodal Models (LMMs) have significantly improved capabilities in visual understanding (Yin et al., 2024), complex instruction following (Wen et al., 2024a; Qin et al., 2024), and multi-step task planning (Li et al., 2024a), enabling a wide range of autonomous agents. Within this landscape, spanning tool-using systems and embodied agents (Fan et al., 2024; Shen et al., 2023), graphical user interface (GUI) agents have emerged as a practical paradigm. By

perceiving screen contents and executing structured actions, GUI agents can operate existing software environments (Liu et al., 2024b; Zhang et al., 2024a; Hu et al., 2025). In particular, mobile GUI agents (Chai et al., 2025; Jiang et al., 2025) have received growing attention due to their wide relevance to common mobile interactions and the ubiquity of smartphones, motivating increasing efforts to automate routine tasks through simulated user interactions (Ye et al., 2025; Tang et al., 2025).

Systematic evaluation of mobile GUI agents is essential for assessing current capabilities and exposing limitations. Existing benchmarks, however, typically focus on perceptual grounding and linear instruction execution. Early work emphasizes GUI element grounding and visual understanding (Li et al., 2025), while later benchmarks extend to single-step or short multi-step instruction following (Rawles et al., 2023; Zhang et al., 2024b), and increasing instruction length to probe task difficulty (Li et al., 2024b). Other efforts explore cross-app interactions (Lu et al., 2024) or broaden coverage to non-English interfaces and localized application ecosystems (Zhang et al., 2025c).

Despite these advances, current mobile GUI benchmarks still exhibit three major limitations. 1) **Limited instruction diversity.** Most benchmarks expand tasks through template substitution or linear concatenation of short instructions (Chen et al., 2024; Lu et al., 2024). Such construction often yields semantically repetitive tasks and reflects biases by a small set of annotators’ brainstorming. In addition, action descriptions often reuse similar phrasing and constraints, limiting the coverage of user’s instruction diversity. 2) **Overly sequential workflows.** Existing datasets typically formalize tasks as linear step-by-step chains with minimal constraints, overlooking decision-aware behaviors involving conditional branching, fallback strategies, or multi-condition filtering. In practice, users commonly adapt their interaction strategies based on

*Equal Contribution. † Corresponding authors.



(a) Template-based and Sequential Tasks

(b) Ours: Complex Tasks with Decision Making and Multiple Constraints

Figure 1: Illustration of conventional step-based tasks with linear chains versus our proposed graph-structured tasks incorporating decision-making and branching conditions.

intermediate outcomes (e.g., “*purchasing a high-priced item if the cheaper one is out of stock*”, or “*adjusting travel plans based on weather forecasts*”). 3) **Limited evaluation protocols.** Current metrics primarily assess step-level correctness or overall task success. While effective for short tasks, these measures do not adequately reflect progress in longer workflows or accuracy at decision points as shown in Fig 1. These limitations hinder systematic evaluation of mobile GUI agents in long-horizon, decision-aware, and cross-context scenarios.

To address these limitations, we introduce **DAC-Bench**, a benchmark for evaluating mobile GUI agents on long-horizon and decision-aware tasks. We develop a human-LLM collaborative **atomic instruction and constraint pipeline** that composes diverse atomic actions into **Graph-structured Compositional Tasks** spanning Sequential, Conjunctive, Conditional and Hierarchical structures. A mixture-of-generators construction strategy with strict manual quality control ensures task executability and semantic diversity. DAC-Bench consists of 830 task episodes with 11,345 actions steps across 35 applications on both Android and iOS platforms, with an average task depth of 13.3 steps. To complement existing evaluation protocols, we introduce **decision-sensitive metrics** that weight long-horizon progress and explicitly assess branching accuracy, enabling finer-grained diagnosis beyond step-level exact match or overall success rate. These components make DAC-Bench a challenging benchmark for identifying agents limitations in **compositional, decision-aware settings that better approximate real-world mobile interaction**

patterns. Our key contributions are as follows:

1. We release **DAC-Bench**, the first benchmark for long-horizon, decision-aware mobile GUI tasks motivated by real-world user interaction patterns.
2. **Principled construction pipeline:** we design a collaborative pipeline that expands atomic instructions with constraints and composes them into graph-structured compositional tasks, balancing executability and semantic diversity.
3. **Decision-sensitive evaluation:** we augment standard step-level and success-based metrics with complementary measures that weight long-horizon steps and explicitly evaluate branching accuracy.
4. **Comprehensive evaluation:** we conduct systematic experiments across 7 general-purpose, mobile-specialized, and reasoning agents, providing diagnostic insights into model limitations.

2 DAC-Bench

2.1 Task Formulation

The **DAC-Bench** dataset consists of a set of k mobile GUI task episodes $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ generated from m candidate mobile applications $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$. As illustrated in Fig 2, for each application A_i , we define an *atomic instruction* set $\{(i_n, c_n) | A_i\}_{n=1}^N$, where i_n represents an available instruction for this application and c_n specifies its associated constraint (if applicable). Following prior work (Lu et al., 2024; Rawles et al., 2023; Li et al., 2024b), each task T_j is represented as a sequence of GUI interaction episodes:

$$T_j = \{(I, i_t, c_t, g_t, a_t)\}_{t=1}^T, \quad (1)$$

where I denotes the global task instruction, i_t is the atomic instruction for step t , c_t is the constraint, g_t is the current screenshot, and a_t is the executed GUI action. Notably, an atomic instruction i_t may internally correspond to multiple low-level actions and screenshots, while still being treated as a single high-level instruction in our formulation.

2.2 Atomic Instruction and Constraints

Prior mobile GUI benchmarks often scale task instructions through template substitution or linearly concatenation, which limits semantic diversity and overlooks application-specific constraints (e.g., “*share via social media*” is a generic action reused across apps regardless of their distinct functionalities). To address these limitations, we design a structured, multi-stage pipeline that integrates human knowledge, LLMs augmentation and rigorous filtering. As illustrated in Fig 2(a), the pipeline consists of: 1) **Human Seeding**: annotators curate a seed set of executable atomic instructions for each application A_i , capturing its core user interactions. 2) **LLM-Augmentation**: to expand beyond the handcrafted set, multiple LLMs (e.g., GPT-4, Claude, DeepSeek) are prompted with the application name, app-store description and seed instructions to iteratively generate additional atomic instructions aligned with the app’s functionality. 3) **Constraint Induction**: for each atomic instruction, LLMs propose up to 3 candidate constraints grounded in application context. Constraints are categorized into a compact taxonomy, including numeric ranges (e.g., “*price within budget*”), thresholds (e.g., “*rating* \geq 4.5”), boolean attributes (e.g., “*free shipping*”), temporal conditions (e.g., “*set the event start time to 3:00 p.m.*”) and preference-based filters (e.g., “*find news related to Sports*”). 4) **Filtering and Verification**: automated LLM-based pre-filtering removes infeasible or redundant outputs, followed by manual verification to ensure executability, realism and diversity. Instructions that differ only by trivial slot-filling entities or minimal lexical variation are discarded (e.g., “*search for Sports news*” and “*find news related to Technology*” are treated as duplicates at the atomic instruction level), and only semantically distinct instructions are accepted. This pipeline produces a diverse library of atomic instructions enriched with application-specific constraints, laying a solid foundation for constructing subsequent graph-structured compositional tasks.

2.3 Compositional Tasks Construction

Building on the atomic instruction library, we construct **Graph-structured Compositional Tasks** that better approximate common mobile interaction patterns. Each task is internally represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $v \in \mathcal{V}$ are *atomic instructions* and edges $e \in \mathcal{E}$ are *constraints* that specify valid transitions. These constraints include the application-specific conditions introduced in Section 2.2, GUI state (e.g., stock availability, rating), contextual signals (e.g, time, weather), as well as necessary predicates or dependencies that govern task progression.

Graph Structures. As illustrated in Fig 2(b), we instantiate four compositional task structures for **atomic-instruction-centric** mobile GUI tasks: 1) **Conjunctive**: multiple sub-constraints that must all be satisfied, with flexible execution order. 2) **Sequential**: a chain of atomic instructions forming a long-horizon task. 3) **Conditional**: a task with a binary decision node and two alternative branches (if/else). 4) **Hierarchical**: a multi-level decision structure formed by nesting conditional branches, simulating user’s fallback and adaptive behaviors.

Task Construction. For each application A_i with its atomic instruction set $\{(i_n, c_n) | A_i\}_{n=1}^N$, we first sample a graph structure together with desired task length and branching depth. A reasoning LLM (Team et al., 2023; Anthropic, 2025; Liu et al., 2024a) is then prompted to select and bind atomic instruction nodes, synthesize constraints for edges, and generate a coherent global natural-language instruction I that describes the overall task.

Quality Control. We observe that different LLMs exhibit complementary behaviors during task construction: GPTs (Achiam et al., 2023) excel at coherent composition but may introduce overly rigid or unnatural decision rules (e.g., “*follow a post if likes are odd, else comment*”); Deepseek (Liu et al., 2024a) produces detailed rationales but often yields shallow rearrangements lacking sequential logic; Gemini (Team et al., 2023) generates more consistent sequential and branching workflows with realistic decision rules. To improve task diversity and robustness, we adopt a **mixture-of-generators strategy** that samples from multiple LLMs during task construction. All generated candidate tasks are then screened through LLM-based checking for feasibility and coherence, followed by human verification to remove trivial variations and ensure semantic diversity. The

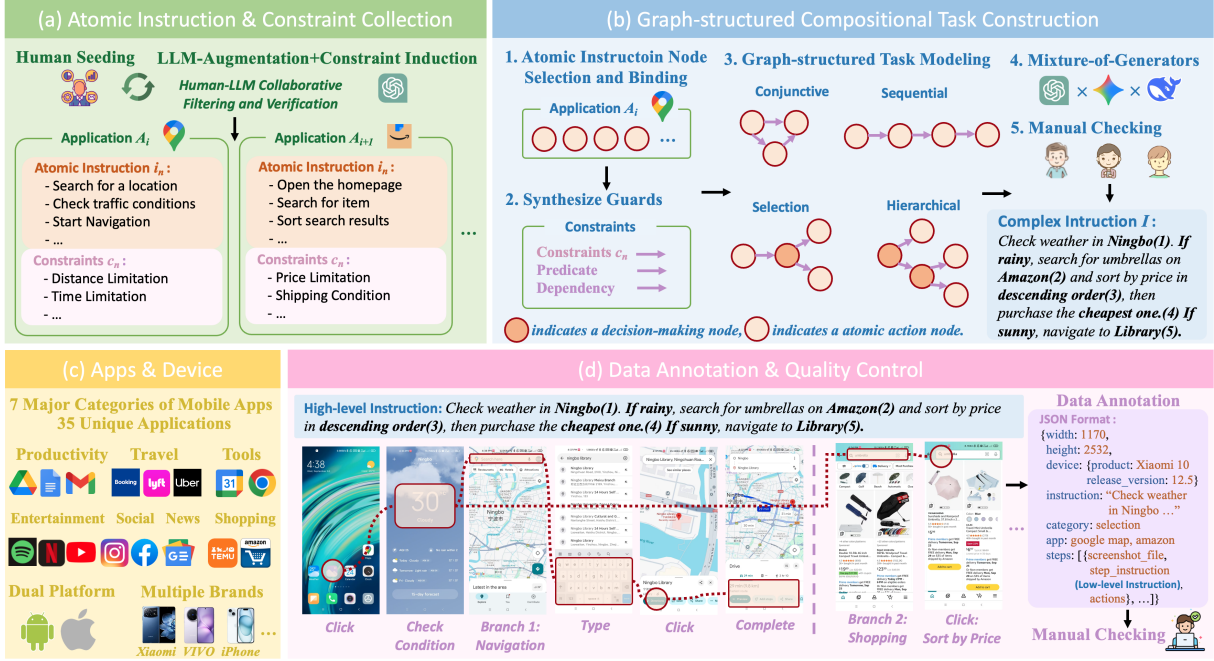


Figure 2: Illustration of DAC-Bench construction pipeline.

resulting tasks yields a curated set of decision-aware, compositional task instructions that form the core of DAC-Bench and support fine-grained evaluation in subsequent experiments.

2.4 Dataset Annotation

Setup. We employed 20 trained annotators to label compositional mobile GUI task instructions. Each annotator was provided with the complete **high-level** instruction and asked to decompose it into step-by-step actions. At each step, annotators produce a corresponding **low-level** (step-level) instruction, describing the specific action taken on the current screen. Following the protocol of prior work (Li et al., 2024b; Lu et al., 2024), this schema enables separate evaluation of agent performance at both high-level and low-level granularity. Annotations were conducted on Android and iOS devices using physical phones, and interactions are recorded through a customized annotation tool that exports data in a unified JSON format.

Annotation. For each step within a task episode, annotators recorded the screenshot, low-level (step-level) instruction, action type, and action parameters. Action types were restricted to a predefined set: {Click, Scroll, Type, Navigate to Home, Navigate to Previous Page, Long Press, Complete}. Metadata including episode IDs, device information, and dimensions were automatically logged. More details are in Appendix C.

Quality Control. To ensure realism, annotators may flag a task as “IMPOSSIBLE” if it violates application usage conventions or could not be completed as instructed. All annotated data then underwent a multi-stage review process: two independent proofreaders and co-authors cross-checked all entries to correct residual errors and verify task consistency and executability.

2.5 Dataset Statistics

We summarize key statistics of DAC-Bench in Tab 1. The dataset contains 830 complex task episodes with a total of 11,345 actions steps, covering 35 widely used mobile applications across 7 categories (Productivity, Travel, Tools, Entertainment, Social Networking, News, Shopping and Payment). For instruction complexity, the average high-level instruction length is 32.54 tokens. Each task contains on average 3.5 atomic instructions and 4 associated constraints, with an average execution depth of 13.3 steps. For task diversity, DAC-Bench includes a library of 2,665 atomic actions and 3,037 constraints, which are composed into the four task structures: Conjunctive (24.3%), Sequential (33.4%), Conditional (24.0%), and Hierarchical (18.3%). In addition, 12.5% of tasks involve cross-app interactions. We further include 250 single atomic instruction tasks as an ablation control group to assess dataset quality and difficulty relative to prior benchmarks. For platform

Dataset	Unique Inst.	Apps	Avg. Steps	High-level & Low-level?	GT	Decision?	Platform	Cross-App?
PixelHelp (Li et al., 2020)	187	4	4.2	high-level	✓		Single	
MoTIF (Burns et al., 2021)	276	125	4.5	high-level & low-level	✓		Single	
AITW (Rawles et al., 2023)	30,378	357	6.5	high-level	✓		Single	
AITZ (Zhang et al., 2024b)	2,504	70	7.5	high-level & low-level	✓		Single	
AndroidControl (Li et al., 2024b)	15,283	833	4.8	high-level & low-level	✓		Single	
AMEX (Chai et al., 2024)	2946	110	12.8	high-level	✓		Single	
GUI Odyssey (Lu et al., 2024)	8,334	212	15.3	high-level & low-level	✓		Single	✓
AppAgent (Zhang et al., 2025a)	45	10	/	high-level			Single	
MobileAgentBench (Wang et al., 2024)	101	10	/	high-level			Single	
Meta-GUI (Sun et al., 2022)	1,125	11	5.3	high-level	✓		Single	
AndroidWorld (Rawles et al., 2024)	/	20	/	high-level			Single	✓
Ours	830	35	13.3	high-level & low-level	✓	✓	Android & iOS	✓

Table 1: Comparison of GUI agent benchmarks

coverage, dataset comprises 881 tasks on Android (version 11 to 15) and 199 tasks on iOS (version 18.5). Among all tasks, 14.1% are available on both systems, while the remainder are platform-specific. Data collection spans 8 phone brands and 16 device models. Tab 1 further compares DAC-Bench with prior datasets across various dimensions.

2.6 Evaluation Metric

Prior mobile GUI benchmarks commonly adopt evaluation metrics such as **Exact Match** (EM), **Type Match** (TM), **Success Rate** (SR) and **Goal Progress** (GP). EM (Li et al., 2024b) requires both agent’s predicted action type and parameters to match the ground truth, while TM only checks the action type. SR measures whether all steps in an episode are executed correctly, but collapses into a binary outcome that is often very low for long instructions and fails to indicate which specific actions caused failure. GP measures the fraction of consecutive correct steps from the beginning of the task, but cannot capture branching structures.

While effective for short, sequential tasks, these metrics provide limited resolution for evaluating long-horizon progress and decision-making at branching nodes, which are common in our compositional mobile GUI tasks. To complement existing evaluation protocols, we introduce two additional metrics that build upon standard step-level measures. **Weighted Longest Common Subsequence (W-LCS)**. Given a predicted action sequence $\hat{\mathcal{T}}$ and the gold sequence \mathcal{T}^* of length $N = |\mathcal{T}^*|$, we compute the LCS between $\hat{\mathcal{T}}$ and \mathcal{T}^* . Let \mathcal{J} denote the set of (*1-indexed*) step positions in \mathcal{T}^* that are matched by the LCS. We define:

$$\text{W-LCS}(\hat{\mathcal{T}}, \mathcal{T}^*) = \frac{\sum_{j \in \mathcal{J}} j}{\sum_{j=1}^N j}. \quad (2)$$

This weighting assigns higher importance to later steps and yields a length-normalized progress score

in $[0, 1]$, reflecting the increased difficulty of maintaining correctness in long-horizon execution. For tasks with conditional nodes, we additionally evaluate **Decision Accuracy**, which measures whether an agent selects the correct branch when encountering a decision constraint. This metric provides a complementary view of an agent’s decision-making capability in non-linear task structures beyond step-by-step execution.

3 Experiments

3.1 Experiment Setting

We comprehensively evaluate a set of mobile GUI agents on DAC-Bench spanning 4 categories: 1) **General-purpose GUI agents** trained on mixed desktop, web and mobile environments, including AGUVIS-7B (Xu et al., 2024), UI-TARS-7B (Qin et al., 2025), Qwen2.5-VL-7B (Bai et al., 2025), OS-Atlas-Pro-7B (Wu et al., 2024); 2) **Mobile-specialized agents** optimized for mobile GUI interactions, AgentCPM-GUI-8B (Zhang et al., 2025c); 3) **Reasoning agents** incorporating explicit reasoning process, InfiGUI-R1-3B and InfiGUI-R1-3B (thinking) (Liu et al., 2025b); 4) **General LMMs** including GPT-5.1, GPT-4o, Claude-sonnet-4.5 and Gemini 3 Pro Preview. Following established evaluation protocols (Zhang et al., 2025c; Lu et al., 2024), each agent is provided with the global high-level instruction I , low-level (step-level) instruction i_t , constraint c_t (if applicable) and corresponding screenshot g_t at each step t . Experiments are conducted using each agent’s official implementation to ensure fairness and reproducibility. Since the action type sets supported by each agent vary, we apply a unified action mapping to the predefined set in Section 2.4 to enable consistent evaluation.

Model	Level	Single				Conjunctive				Sequential				Conditional				Hierarchical			
		EM	TM	SR	GP	EM	TM	SR	GP	EM	TM	SR	GP	EM	TM	SR	GP	EM	TM	SR	GP
AGUVIS-7B	LL	87.81	95.55	72.40	72.47	88.75	97.56	50.00	61.39	90.70	97.87	41.88	59.55	90.51	97.07	49.25	72.46	91.30	98.32	33.55	67.18
	HL	42.59	59.44	32.00	26.91	33.33	60.01	3.48	11.75	26.71	49.18	0.36	11.86	22.49	41.27	0.50	16.83	21.54	42.38	0.66	12.42
UI-TARS-7B	LL	87.11	97.20	63.60	70.03	79.51	97.37	26.24	44.61	80.72	96.62	24.55	43.76	85.33	96.18	26.13	59.11	83.36	97.51	15.79	49.79
	HL	63.34	79.56	31.60	35.68	53.97	78.25	3.96	14.49	55.03	79.31	2.53	16.48	48.63	69.64	1.01	18.71	45.93	71.87	0.66	15.39
Qwen2.5-VL-7B	LL	79.90	91.21	62.80	59.99	81.26	92.98	36.14	51.76	76.62	87.94	24.19	46.25	80.44	91.65	19.10	51.61	75.68	86.08	11.18	48.32
	HL	58.55	73.53	38.00	35.66	53.66	71.43	5.45	17.45	44.80	59.01	4.33	18.70	41.82	57.53	1.01	19.54	36.31	50.99	0.66	16.02
OS-Atlas-Pro-7B	LL	80.99	90.94	67.60	67.32	74.94	90.02	31.19	47.03	67.10	83.76	18.05	39.53	68.51	82.75	16.58	46.34	60.98	79.74	4.61	36.78
	HL	54.31	70.38	36.40	33.12	46.21	73.38	6.44	16.25	47.34	72.09	1.44	12.95	40.03	65.04	0.00	14.10	37.57	65.60	0.00	10.96
AgentCPM-GUI-8B	LL	83.85	90.14	68.80	69.05	84.20	93.26	41.09	54.09	86.77	93.91	28.16	49.82	86.95	92.42	33.17	61.06	87.74	93.26	19.08	57.58
	HL	62.48	75.82	32.40	35.73	57.94	78.14	8.91	21.53	55.97	78.84	2.53	16.90	50.39	71.23	1.01	18.15	47.28	73.03	0.66	15.00
InfiGUI-R1-3B	LL	58.26	71.98	59.60	46.93	61.48	75.09	25.74	38.57	62.81	80.26	13.00	35.58	57.87	74.71	9.55	38.29	59.18	77.81	3.95	32.00
	HL	37.58	50.63	27.20	24.11	38.03	57.39	0.99	10.19	42.73	63.38	1.08	8.91	37.07	56.30	0.00	10.49	38.54	62.58	0.66	9.75
InfiGUI-R1-3B tk	LL	60.97	74.11	65.60	49.28	69.78	84.74	34.65	44.29	69.77	89.80	22.74	43.14	63.57	83.80	17.09	44.20	66.53	88.53	9.87	41.52
	HL	44.35	63.29	35.60	28.77	44.99	70.57	4.95	14.08	49.06	73.32	1.81	12.59	40.28	63.82	2.01	16.21	41.97	68.08	0.00	12.22
GPT-5.1	LL	48.60	93.80	11.60	20.25	39.41	93.45	1.98	12.24	42.24	95.17	2.17	10.14	46.70	94.29	0.50	13.35	44.28	94.26	0.00	9.92
	HL	38.43	80.54	6.00	12.76	32.00	80.81	0.00	6.16	32.86	80.16	0.00	4.52	34.24	72.66	0.00	8.36	33.07	75.11	0.00	4.73
GPT-4o	LL	36.65	86.14	6.00	11.23	30.83	88.35	0.99	5.12	29.64	85.26	0.00	4.48	30.14	81.56	0.00	4.93	33.06	82.46	0.00	6.04
	HL	29.23	68.68	3.20	7.35	25.15	76.27	0.00	3.42	26.26	75.13	1.08	3.31	25.11	66.94	0.00	3.79	24.45	68.95	0.00	3.30
Claude-sonnet-4.5	LL	37.43	88.84	10.04	5.58	27.66	89.72	0.99	4.97	27.57	91.61	1.44	3.53	31.65	91.63	0.50	4.44	30.19	91.39	0.00	3.18
	HL	27.08	77.96	5.67	3.71	21.93	81.17	0.99	3.27	23.36	80.96	0.36	2.33	24.13	75.54	0.00	4.39	22.37	76.60	0.00	2.60
Gemini 3 Pro Preview	LL	57.13	90.82	18.00	28.36	52.04	93.95	3.96	15.92	56.94	95.09	0.72	12.18	58.90	93.94	0.50	17.49	56.63	93.50	0.00	13.02
	HL	51.86	82.71	18.40	27.11	40.62	83.45	0.99	9.37	43.67	81.95	0.36	7.54	42.39	78.66	0.00	12.18	39.13	79.30	0.00	7.34

Table 2: Main results for mobile GUI agents. “InfiGUI-R1-3B tk” indicates the thinking mode for InfiGUI-R1-3B.

3.2 Main Results

Tab 2 summarizes the overall performance of evaluated agents on DAC-Bench. We observe a clear difficulty gradient: both SR and GP decrease as task structures become more complex. On the most complex Hierarchical subset, the average SR falls to 14.00% even under low-level setting, and GP averages merely 13.10% under high-level setting. Among evaluated agents, AGUVIS-7B achieves the highest 89.81% overall EM under low-level setting and AgentCPM-GUI-7B with the highest 53.31% EM under high-level setting. Notably, InfiGUI-R1-3B-thinking exhibits the smallest degradation (2.38%) from Conjunctive/Sequential to Conditional/Hierarchical, indicating strongest robustness to task length and structural variation.

Comparison with prior benchmarks. To verify that the increased difficulty of DAC-Bench is not an artifact of data construction (e.g., artificially extended step descriptions or redundant GUI elements), we include a Single Atomic Instruction subset as control group. Agent performance on this subset is comparable to that on prior single-step benchmarks (81.19% on ours v.s. 80.81% on AndroidControl (Li et al., 2024b), a widely used mobile GUI benchmark), confirming that DAC-Bench preserves the feasibility of basic action execution.

To further examine differences beyond task statistics alone, we compare agent performance on DAC-Bench with that on widely used mobile GUI benchmarks. Averaged across task structures, DAC-Bench results in an accuracy drop of 20.69% relative to prior datasets. In contrast to sequential-only

benchmarks such as AndroidControl, DAC-Bench places greater emphasis on structural correctness: under high-level instructions, EM is 24.69% lower than AndroidControl, and even AndroidControl’s hardest split remains 26.29% higher than the Conditional/Hierarchical subsets of DAC-Bench. These results indicate that DAC-Bench introduces additional difficulty from **long-horizon, decision-aware task structures**, while maintaining comparable performance on atomic instruction execution.

3.3 Ablation Studies

High-level v.s. Low-level Instructions. Tab 2 compares agent performance under low-level (step-specified) and high-level (goal-only) instructions. Across all agents, EM/TM are consistently higher with low-level guidance, indicating that most agents can reliably execute explicitly grounded actions. When shifted to high-level instructions, TM drops by 24.11% on average and EM falls below 32.78%, revealing a gap between current agents’ strong visual grounding/navigation and weak global task reasoning. In real-world scenarios, users rarely give perfectly disambiguated step instructions making this gap even more apparent as task complexity increases.

Impact of Task Length. Under low-level setting (Tab. 2), TM/EM remains relatively stable as task length increases, since each step is explicitly specified. In contrast, high-level setting exhibit strong length sensitivity (Fig. 3): GP decreases sharply with longer sequences, falling below 20% once tasks exceed 6-8 steps. This trend highlights

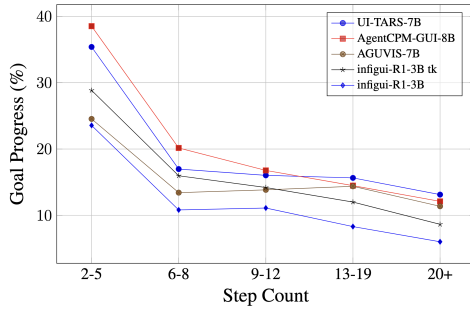


Figure 3: High-level GP v.s. Step Length.

cumulative error propagation in long execution.

To better capture partial progress in such settings, we report W-LCS, which measures how far an agent proceeds before failure. As shown in Fig 4, AgentCPM-GUI-8B maintains higher W-LCS as task length increases, even its GP/SR are not the highest in Tab 2. InfiGUI-R1-3B (thinking) exhibits only moderate W-LCS under low-level settings, but shows a clear advantage on tasks longer than 13 steps under high-level instructions. This suggests their superior resistance to error accumulation and greater robustness in complex reasoning. In contrast, AGUVIS-7B performs strongly under low-level settings but degrades substantially under high-level instructions, suggesting limited robustness to long-horizon planning.

Effect of Branching Structures. Conditional/Hierarchical tasks introduce decision nodes where a single incorrect choice invalidates subsequent execution, leading to sharper performance drops than Sequential/Conjunctive tasks. We quantify this effect using decision accuracy by branch depth in Tab 3. The first decision node achieves an average accuracy of only 36.85%, with deeper nodes in Hierarchical tasks decreasing by a further 7.08% on average. Among all agents, the InfiGUI-R1-3B series, a smaller model series with reasoning SFT and RL, outperforms other agents by 3.89% on first decision and 10.12% on deeper nodes, suggesting **reasoning-oriented training enhances decision robustness**, a property largely absent in baseline agents. We conducted a supplementary experiment on GRPO finetuning Qwen2.5-VL with 200 DAC-Bench decision nodes and evaluate on whole dataset. We noticed that GRPO finetuning on these decision-aware actions, increase the first branch decision accuracy by 3.46% and subsequent accuracy by 8.47%. Overall, these results indicate that decision-making under branching structures remains a challenge for mobile GUI agents.

Model	First Decision	Deeper Decisions	EM Difference
AGUVIS-7B	31.29%	29.11%	-2.18%
UI-TARS-7B	45.85%	29.72%	-16.13%
Qwen2.5-VL-7B	37.46%	30.56%	-6.90%
OS-Atlas-Pro-7B	32.38%	29.72%	-2.66%
AgentCPM-GUI-8B	37.25%	29.72%	-7.53%
InfiGUI-R1-3B	40.17%	37.95%	-2.22%
InfiGUI-R1-3B tk	41.31%	41.83%	+0.52%

Table 3: Decision accuracy (EM \uparrow) at different branching depths under high-level instructions. “-” denotes accuracy degradation on deeper nodes.

3.4 Case studies and Error Analysis

Q1. Why do agents fail on complex tasks?

We analyze representative failure cases using InfiGUI-R1-3B, which provides access to intermediate reasoning traces. We observe that agents’ failure modes fall into 2 major categories: **perception-related errors** and **reasoning-related errors**.

Perception-related errors reveal agents’ **visual grounding limitations**, rather than purely “reading” failures caused by Optical Character Recognition (OCR). Notably, in many cases agents can correctly read decision-relevant signals (numbers, labels, toggle states), but still fail to translate them into the correct interaction. Four main failure modes are observed: **Mode 1: UI affordance misunderstanding**. Agents may misinterpret which UI element affords the intended operation (e.g., clicking the article title instead of “remove from favorites” button). **Mode 2: Off-screen/hidden UI access failures**. Agents sometimes fail to infer that actions such as Scroll/Swipe/Long Press are needed to reveal missing buttons/apps, resulting in repeated interactions with visible but irrelevant components. **Mode 3: Localization and precision errors**. In some cases despite the correct intent, the action type is correct but localization is imprecise (e.g., tapping near small or densely packed targets), leading to failures that are not captured by type-level correctness alone. **Mode 4: Termination and final-state inference errors**. Agents may prematurely conclude that a task is complete or fail to recognize completion (e.g., assuming a TV show is already added to “My List” and skipping the subsequent required steps). This often results in not issuing Complete/Stop and entering repetitive loops in long-horizon tasks.

As illustrated in Fig 5, reasoning-related errors reveal agents’ **decision-making weakness**. These errors mainly arise from decision execution rather than generic reasoning difficulty. Two recurring failure modes are observed. **Mode 1: Ignoring branching constraints**. The agent often defaults

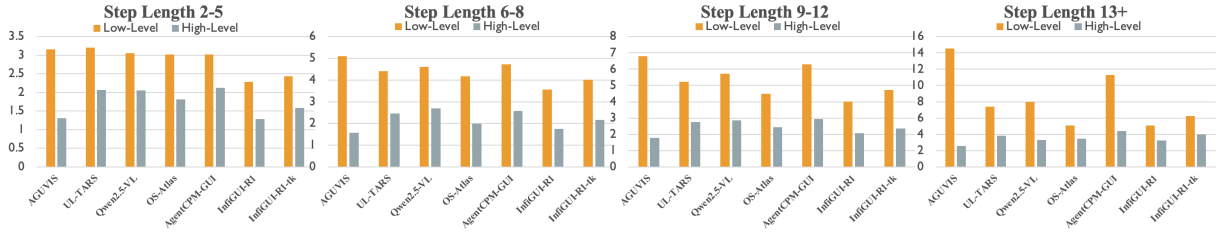


Figure 4: Visualization of W-LCS scores as a function of task step length.

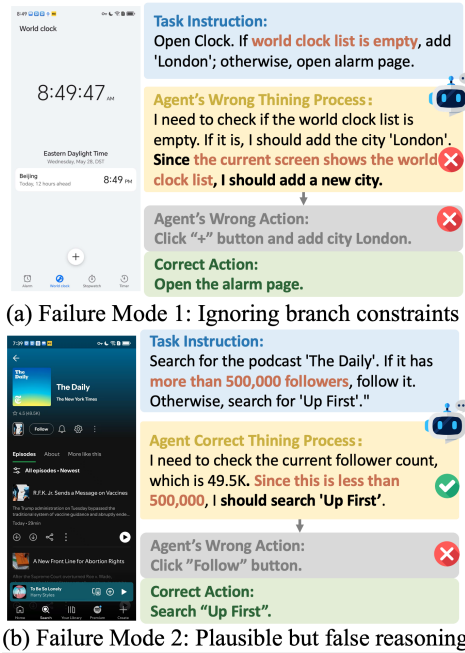


Figure 5: Decision failure cases.

to the first available option without correctly applying the specified decision condition, thereby bypassing the decision step. **Mode 2: Plausible but inconsistent reasoning.** In some cases, the agent produces a coherent reasoning trace that acknowledges both the instruction and GUI state, but subsequently executes an action that contradicts with its own reasoning, again defaulting to the first option. These failure patterns highlight a discrepancy between strong step-level execution and weak conditional decision-making: while agents achieve near-saturated TM/EM on Single, short Sequential and low-level settings, they struggle when required to compare multiple candidate actions against global instructions and on-screen conditions. One contributing factor is the limited exposure of current mobile GUI agents to decision-oriented task structures during training, which constrains their ability to reliably handle branching decisions.

Q2. Besides task length and branching, what other factors limit agent performance?

Model	CLICK		TYPE		SCROLL		Complete		Others	
	TM	EM	TM	EM	TM	EM	TM	EM	TM	EM
Low-level										
AGUVIS-7B	99.84	91.65	95.28	91.89	99.04	87.33	86.85	86.85	93.69	83.95
UI-TARS-7B	99.16	79.01	94.40	89.15	99.00	94.19	99.04	99.04	44.97	44.97
Qwen2.5-VL-7B	93.12	83.21	84.34	81.49	81.01	46.04	89.86	89.86	55.38	53.23
OS-Atlas-Pro-7B	94.96	74.66	85.67	82.16	48.65	28.94	67.31	67.31	68.51	62.66
AgentCPM-GUI-8B	97.27	88.30	89.32	84.98	98.27	96.61	79.79	79.79	40.75	31.51
InfiGUI-R1-3B	87.04	79.14	85.16	82.67	81.81	5.64	10.43	10.43	86.17	22.95
InfiGUI-R1-3B tk	97.94	88.79	89.78	86.84	96.28	7.69	8.91	8.91	90.56	24.21
High-level										
AGUVIS-7B	67.48	34.68	28.69	23.52	14.27	11.02	0.00	0.00	43.83	43.83
UI-TARS-7B	88.45	54.74	64.32	53.35	52.34	47.74	44.81	44.81	28.53	28.53
Qwen2.5-VL-7B	67.41	44.79	53.25	48.00	15.25	11.40	79.66	79.66	18.87	18.87
OS-Atlas-Pro-7B	86.37	48.53	54.50	44.43	29.85	26.71	35.55	35.55	41.61	41.61
AgentCPM-GUI-8B	86.50	54.24	64.67	52.45	45.42	41.83	66.19	66.19	29.56	29.24
InfiGUI-R1-3B	88.56	55.16	57.24	49.33	41.38	23.45	9.05	8.98	36.48	29.87
InfiGUI-R1-3B tk	75.17	46.57	54.22	47.91	32.42	20.24	17.47	17.40	33.65	28.00

Table 4: Model performance on different action types.

Action Type Effects. Tab 4 shows substantial performance variation across action types. Long Press, a realistic yet underrepresented action, exhibits the lowest accuracy across agents (on average 66.31% lower than Click/Type). For UI-TARS and OS-Atlas, the Complete action underperforms other agents by 52.03%, indicating their difficulty in recognizing task termination in long-horizon settings. Their overall SR rises by 8.18% when Complete is excluded, suggesting that many failures stem from termination misrecognition rather than execution errors. Annotators also noted the near absence of Double-Tap: although often replaceable by alternative actions, its omission from most agents’ action space and training data may limit generalization. These observations highlight the impact of action space coverage and final-state inference on agent reliability.

History Window Length. Long-horizon tasks place increased demands on historical context management. We observe that the optimal history length varies across agents. For instance, AgentCPM achieves its highest high-level accuracy with a 4-step history window (1.38% higher than full-history or no-history). However, its low-level accuracy declines as history grows, indicating that longer context is not uniformly beneficial. In contrast, AGUVIS benefits from longer histories under high-level instructions but shows limited sensitivity under low-level settings. We identify this as

a trade-off among multiple factors: 1) **Contextual Overload**: excessively long textual/histories may increase confusion and accumulative error carry-over. 2) **Visual Down-sampling**: aggregating many screenshots reduces effective resolution, harming precision actions such as small UI target clicks. 3) **Efficiency**: longer histories increase inference latency, problematic for real-time mobile scenarios. These results suggest that effective history utilization depends on model design rather than context length alone.

Instruction Language and Platform. We further analyze performance across instruction languages and platforms (more results in Appendix G and F). Qwen2.5-VL-7B demonstrates greater robustness to English–Chinese instruction shifts, even achieving 3.5% higher EM on Chinese instructions. In addition, we observe that InfiGUI-R1-3B series benefits from its “thinking” prompting, effectively reduces outputs formatting errors. Across agents, high-level instructions yield more irregular outputs than low-level ones, and Chinese prompts are generally more error-prone than English. For platform difference, since most mobile GUI training corpora skewed toward Android, average performance drops by 5.30% on iOS, likely due to GUI styling and action differences.

Q3. What actionable directions does DAC-Bench suggest for future agents? Based on the above diagnosis, we suggest several targeted directions for future mobile GUI agents: (i) **decision-node-focused pre-training** to improve branch selection and constraint-to-action binding (consistent with the gains we observe from decision-node-only GRPO), (ii) **rare-action and visibility training** to reliably execute Scroll/Swipe/Long Press and reveal hidden controls, (iii) **localization calibration** for small UI targets under varying resolutions and layouts (especially same app across Android and iOS platforms), and (iv) **improved termination/final-state inference** to reduce repetitive loops and trigger Complete/Stop appropriately.

4 Conclusion

In this paper, we presented **DAC-Bench**, a benchmark designed to evaluate mobile GUI agents on **long-horizon** and **decision-aware** task execution. By constructing compositional tasks with sequential, conjunctive, conditional, and hierarchical structures, DAC-Bench extends beyond prior benchmarks that focus primarily on short, linear

workflows. Experimental results show that, while current agents perform reliably on single-step actions, their performance degrades substantially on longer tasks and branching decisions, revealing limitations not captured by existing evaluation settings. We hope DAC-Bench can serve as a diagnostic benchmark and facilitate future progress in decision-aware mobile GUI agent design.

Acknowledgements

This work was in part supported by National Natural Science Foundation of China (No. 62402429, U24A20326, 62441236), the Key Research and Development Program of Zhejiang Province (No. 2025C01026), Ningbo Yongjiang Talent Introduction Programme (2023A-397-G), Young Elite Scientists Sponsorship Program by CAST (2024QNR001) and Tencent Rhino-Bird Joint Research Program. The authors gratefully acknowledge the support of Zhejiang University Education Foundation Qizhen Scholar Foundation.

Limitation

DAC-Bench follows an offline, static episode-replay paradigm: each task is evaluated on recorded screenshots paired with a human-executed trajectory. This design enables controllable and reproducible comparisons across agents, but it does not fully capture challenges in online interactive environments, such as non-stationary in-app content, system interruptions (notifications/permission prompts), or other stochastic dynamics (in-app advertisement and banners). Online evaluation is a complementary paradigm for testing robustness under deployment-like conditions, and we plan to explore it as a future extension.

Our primary evaluation compares agents against a single reference (gold) trajectory per episode, which may undercount alternative yet valid strategies in real apps (e.g., search-first vs. category navigation). We mitigate severe ambiguity through quality control and report progress-sensitive metrics (e.g., W-LCS) to reduce false negatives from minor deviations, but this limitation is not fully eliminated. As future work, we plan to incorporate multiple reference trajectories (or strategy-level equivalence classes) and complementary goal-state/end-state checks to better account for valid behavioral diversity.

We acknowledge that although DAC-Bench covers diverse application categories, platforms

(Android/iOS), and bilingual instructions (English/Chinese), it remains limited in cultural and linguistic scope, including more device brands and models (e.g., Mobile Tablets and iPads), more application categories and comparison of same application under different language settings (e.g., Chinese applications may contain advertisements compared with their English versions). Future extensions may address these limitations to ensure broader fairness and inclusiveness. We consider actively incorporating more downstream application categories such as Medical, Education and Games.

We also acknowledge that current workflows are human-LLM approximation of real-world user actions, and may not fully reflect all user cases and complex decisions. Additionally, although we allow annotators mark some tasks as “IMPOSSIBLE”, ambiguous or conflicting tasks may still exist in real user requests and our dataset has limited coverage for such scenarios.

Despite employing 20 professional annotators and a dual-review process, a small number of ambiguous or unclear low-level instructions may still exist. The linguistic diversity of high-level instructions remains limited (ambiguous, colloquial phrasing characteristic of real users), with most maintaining relatively standardized expressions.

We did not systematically verify potential overlap between DAC-Bench and other agents’ training data in this work (although our data are unique by paper submission), so we cannot entirely rule out the possibility that a small number of task patterns appeared in other agents’ training.

Ethical Considerations

DAC-Bench was constructed with careful consideration of privacy, safety and fairness. All task instructions and screenshots were collected by professional annotators on mobile devices using publicly available mobile applications and no personal or sensitive user data are included. The dataset contains only synthetic task instructions generated during controlled annotation as stated in Section 2, ensuring that no private information is exposed. No emulators are used for annotation.

All annotators, co-authors and quality checkers are clearly notified the instruction, meta data and screenshot usage. Annotators are paid at market price according to local laws and requirements.

All applications were operated in safe environments without accessing sensitive content such

as payments, contacts, or personal communications (all mentioned user profiles, names, emails and accounts are separately created with no relations to real-world individuals). The dataset is intended solely for academic research on mobile GUI agents and is released under a non-commercial license to discourage misuse.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anthropic. 2025. [Claude-3.7-sonnet](#). Accessed: 2025-09-25.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-v1 technical report. *arXiv preprint arXiv:2502.13923*.
- 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. *arXiv preprint arXiv:2104.08560*.
- Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Shuai Ren, and Hongsheng Li. 2024. Amex: Android multi-annotation expo dataset for mobile gui agents. *arXiv preprint arXiv:2407.17490*.
- Yuxiang Chai, Hanhao Li, Jiayu Zhang, Liang Liu, Guangyi Liu, Guozhi Wang, Shuai Ren, Siyuan Huang, and Hongsheng Li. 2025. A3: Android agent arena for mobile gui agents. *arXiv preprint arXiv:2501.01149*.
- Jingxuan Chen, Derek Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Li Yixing, Xurui Zhou, Weiwen Liu, Shuai Wang, and 1 others. 2024. Spabench: A comprehensive benchmark for smartphone agent evaluation. In *NeurIPS 2024 Workshop on Open-World Agents*.
- Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Workflowllm: Enhancing workflow orchestration capability of large language models. *arXiv preprint arXiv:2411.05451*.
- Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, and 1 others. 2025. Os agents: A survey on mllm-based agents for general computing devices use. *arXiv preprint arXiv:2508.04482*.
- Wenjia Jiang, Yangyang Zhuang, Chenxi Song, Xu Yang, Joey Tianyi Zhou, and Chi Zhang. 2025. Appagentx: Evolving gui agents as proficient smartphone users. *arXiv preprint arXiv:2503.02268*.
- Juyong Lee, Taywon Min, Minyong An, Dongyoon Hahn, Haeone Lee, Changyeon Kim, and Kimin Lee. 2024. Benchmarking mobile device control agents across diverse configurations. *arXiv preprint arXiv:2404.16660*.
- Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*.
- Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, and 1 others. 2024a. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534.
- Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024b. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37:92130–92154.
- Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile ui action sequences. *arXiv preprint arXiv:2005.03776*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Guangyi Liu, Pengxiang Zhao, Liang Liu, Zhiming Chen, Yuxiang Chai, Shuai Ren, Hao Wang, Shibo He, and Wenchao Meng. 2025a. Learnact: Few-shot mobile gui agent with a unified demonstration benchmark. *arXiv preprint arXiv:2504.13805*.
- Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, and 1 others. 2024b. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820*.
- Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. 2025b. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*.
- Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.
- Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. Uitars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

- Christopher Rawles, Sarah Clinckemahillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2024. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.
- Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugging-gpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.
- Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. *arXiv preprint arXiv:2205.11029*.
- Fei Tang, Haolei Xu, Hang Zhang, Siqi Chen, Xingyu Wu, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Zeqi Tan, Yuchen Yan, and 1 others. 2025. A survey on (m) llm-based gui agents. *arXiv preprint arXiv:2504.13865*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. 2024. Mobileagentbench: An efficient and user-friendly benchmark for mobile llm agents. *arXiv preprint arXiv:2406.08184*.
- Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaying Xu, and 1 others. 2024a. Benchmarking complex instruction-following with multiple constraints composition. *Advances in Neural Information Processing Systems*, 37:137610–137645.
- Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024b. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 543–557.
- Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and 1 others. 2024. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.
- 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. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6061–6072.
- Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024. Aguis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.
- 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*.
- Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.
- 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*, pages 1–20.
- 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. *arXiv preprint arXiv:2403.02713*.
- Shaoqing Zhang, Kehai Chen, Zhuosheng Zhang, Rumei Li, Rongxiang Weng, Yang Xiang, Liqiang Nie, and Min Zhang. 2025b. Xbound: Exploring the capability boundaries of device-control agents through trajectory tree exploration. *arXiv preprint arXiv:2505.21279*.
- Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, and 1 others. 2025c. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*.

A Appendix: Related Work

Prior efforts in mobile GUI benchmark construction can be grouped into three following paradigms. 1) Human Handicraft Designed Instructions: AITW (Rawles et al., 2023) categorizes tasks based on the number of action steps into multiple-step and single-step categories. Multiple-step tasks are obtained through human annotation, technical documentation, and LLM-enhanced instructions, while single-step tasks are derived by extracting shorter action sequences from longer action sequences. AITZ (Zhang et al., 2024b), built upon AITW, further filtered out incorrect or mismatched instructions and utilized GPTs to assist in proofreading task instruction semantic descriptions. Other mainstream datasets and benchmarks (Rawles et al., 2024; Sun et al., 2022; Li et al., 2020; Wen et al., 2024b; Lee et al., 2024) widely adopted such human written instructions to simulate real-world user cases by either crowd sourcing or mimicking daily user case and routine. 2) Extracted and generated from large-scale dataset or website (Liu et al., 2025a; Chai et al., 2024; Zhang et al., 2025b): Android-arena (Xing et al., 2024) identify webpage and descriptions related to application functionality through web retrieval, then store these software descriptions in a vector database, and finally utilize an LLM to reconstruct app tasks instructions from them. 3) Expand based on existing instructions and template replacement: GUI-Odyssey (Lu et al., 2024) through manually brainstorming instruction task templates, through template substituting the application name or action description, quickly expand current instruction to a large-scale dataset and finally rewrite with GPT-4. SPA-Bench (Chen et al., 2024) first start with single-step instruction and through expanding one action at a time to obtain long chain instruction and finally achieve difficulty rising. Despite these efforts, existing benchmarks exhibit limitations: heavy reliance on human annotation or template substitution, limited instruction diversity, and a bias toward chain-like workflows. Consequently, they fall short of simulating the decision-making and judgment-based behaviors seen in real user operations, which highlights our design principles.

B Appendix: Use of LLMs

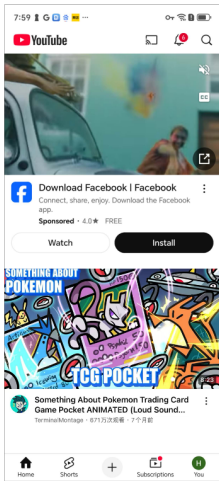
Our LLM usage are stated in Section 2 and concluded as below. *Instruction Augmentation*: we utilized multiple LLMs for expanding human seed atomic instructions, collecting constraints and iteratively refinement with human collaboration. *Complex Instruction Construction*: we utilized multiple LLMs for construction longer instructions from atomic actions, iterative enhancing and quality checking (e.g., grammar, coherence, feasibility). We ensured that all LLM generated instructions are double verified by either paper co-authors or annotators to prohibit usage/inappropriate instructions, with essential manual re-writing.

The experiments related to mobile GUI agents are conducted with LLMs, all models (checkpoints, configurations and licenses) are obtained from their official open-source implementation without additional training or modification. We did not use LLM-generated contents for paper writing, but merely grammar checking.

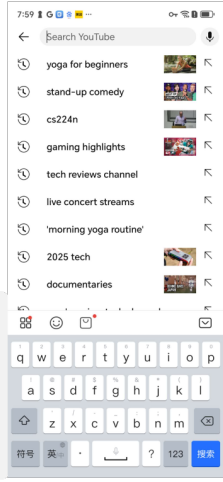
C Appendix: Annotation Details

We wrote a standardized annotation guideline for the 20 annotators, with example JSON file format, example annotated data and other required information. Here we provide a snapshot of our proposed dataset meta data, with more details available in code base.

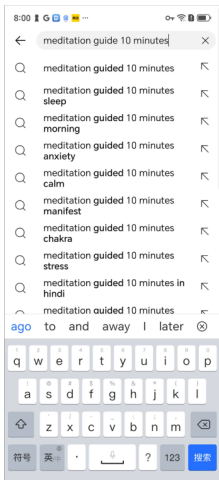
Meta Data :
 {episode_id: 132,
 width: 1170,
 height: 2532,
 device: {product: Honor X60 Pro
 release_version: 14,
 sdk_version: 34}
 instruction: "First, search YouTube for 'meditation guide 10 minutes' and play the first search result. If the video title contains 'beginner', then create a new note in Evernote titled 'Meditation Resources'. Otherwise, add the video to your 'Watch Later' playlist on YouTube."
 category: selection
 app: [Youtube, Evernote]



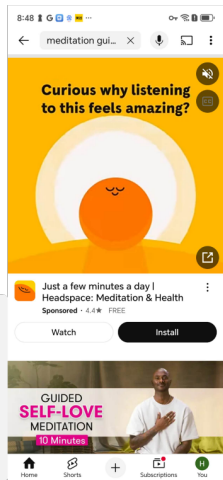
Step Instruction :
 {screenshot: ./data/screenshots/132/132_0.jpg,
 step_instruction: click search button to search,
 actions: [{action_type: click,
 location: [1091,160,1204,242]}]}



Step Instruction :
 {screenshot: ./data/screenshots/132/132_2.jpg,
 step_instruction: click blue search at bottom
 actions: [{action_type: click,
 location: [663,1268,894,1373]}]}



Step Instruction :
 {screenshot: ./data/screenshots/132/132_1.jpg,
 step_instruction: type 'meditation guide 10 minutes' into the search,
 actions: [{action_type: type,
 text: meditation guide 10 minute}]}



Step Instruction :
 {screenshot: ./data/screenshots/132/132_2.jpg,
 step_instruction: scroll down to watch title,
 actions: [{action_type: scroll,
 orientation: down}]}

Figure 6: Illustration of annotation details and meta data.

D Appendix: Step Length Experiments

Here we provide detailed experimental Results of Agent’s performance across different step length in compensation to Section 3.3 discussions. Here 1a denotes atomic action level GP, 1b denotes atomic action level GP with branching length weighted. 2a denotes step level GP, 1b denotes step level GP with branching length weighted. 3a denotes the longest common sequence from first step, 3b denotes the longest common sequence from first step with branching length weighted. 4a denotes the longest common sequence taking all steps into consideration, 4b denotes the longest common sequence taking all steps into consideration with branching length weighted. 5a denotes the longest common sequence taking all steps into consideration with task length weighted, 5b denotes the longest common sequence taking all steps into consideration with branching length and task length weighted. 6a denotes the longest common sequence percentage, 6b denotes the longest common sequence percentage with task length weighted.

Model	Group	Level	Tasks	1a (%)	1b (%)	2a (%)	2b (%)	3a (Abs)	3b (Abs)	4a (Abs)	4b (Abs)	5a (Abs)	5b (Abs)	6a (%)	6b (%)
AGUVIS	2-5 steps	HL	281	1.48	2.76	0.85	0.95	0.03	0.06	1.20	1.31	0.81	0.84	24.54	22.63
		LL	281	63.32	62.72	62.56	61.28	0.75	0.89	2.90	3.16	2.55	2.75	72.53	71.39
	6-8 steps	HL	225	6.35	8.82	4.12	4.17	0.17	0.24	1.58	1.58	0.85	0.84	13.43	13.11
		LL	226	54.03	58.12	52.00	52.12	1.15	1.51	5.02	5.09	3.96	4.02	62.02	61.91
	9-12 steps	HL	222	10.79	12.32	6.27	6.07	0.28	0.34	1.77	1.79	1.13	1.14	13.85	13.58
		LL	222	56.73	59.54	53.94	53.57	1.42	1.71	6.67	6.79	5.49	5.58	64.77	64.51
	13+ steps	HL	351	10.49	10.99	5.94	5.16	0.33	0.38	2.51	2.57	1.76	1.80	13.16	11.80
		LL	351	59.32	59.82	55.16	55.05	1.91	2.14	11.98	14.56	9.78	11.79	65.16	64.89
	Overall	HL	1079	7.34	9.94	4.30	4.85	0.21	0.31	1.82	2.18	1.19	1.47	16.32	13.23
		LL	1080	58.72	59.81	56.17	54.91	1.35	1.79	7.07	10.88	5.80	8.82	66.34	64.98
AgentCPM	2-5 steps	HL	281	28.20	27.40	27.92	23.97	0.33	0.37	1.99	2.12	1.25	1.27	38.52	34.99
		LL	281	53.05	55.01	52.57	51.07	0.65	0.81	2.77	3.02	2.44	2.64	69.79	68.59
	6-8 steps	HL	226	13.09	16.46	11.42	11.42	0.32	0.46	2.56	2.58	1.27	1.27	20.16	19.83
		LL	226	50.21	54.25	48.10	47.65	1.08	1.42	4.66	4.72	3.73	3.75	58.90	58.39
	9-12 steps	HL	222	12.57	15.37	8.95	8.56	0.34	0.45	2.91	2.94	1.37	1.37	16.78	16.29
		LL	222	50.90	56.91	47.13	46.87	1.32	1.67	6.19	6.30	4.78	4.88	57.45	57.36
	13+ steps	HL	351	10.85	11.49	6.62	5.92	0.37	0.42	4.01	4.40	1.89	2.03	13.52	12.37
		LL	351	45.50	47.52	39.48	38.77	1.52	1.75	9.49	11.31	7.22	8.64	49.05	48.28
	Overall	HL	1080	16.19	15.06	13.65	8.67	0.34	0.43	2.96	3.69	1.49	1.74	22.09	16.00
		LL	1080	49.56	51.60	46.26	42.52	1.16	1.56	6.05	8.79	4.75	6.78	58.23	53.05
UI-TARS	2-5 steps	HL	281	25.86	24.85	25.28	20.95	0.30	0.34	1.94	2.07	1.14	1.14	35.39	31.66
		LL	281	69.04	69.48	68.50	64.83	0.83	1.02	2.98	3.21	2.50	2.65	72.95	70.05
	6-8 steps	HL	226	9.28	12.19	7.28	7.46	0.23	0.33	2.43	2.45	1.08	1.10	16.98	16.92
		LL	226	43.74	49.78	41.75	41.28	0.97	1.27	4.37	4.41	3.51	3.52	56.16	55.55
	9-12 steps	HL	222	12.08	14.39	7.77	7.43	0.33	0.42	2.73	2.76	1.32	1.33	16.03	15.73
		LL	222	43.06	49.36	38.91	38.33	1.10	1.38	5.14	5.21	4.20	4.27	52.47	51.99
	13+ steps	HL	351	11.99	12.53	7.06	6.22	0.40	0.44	3.53	3.80	2.02	2.23	14.63	13.46
		LL	351	32.41	34.10	25.44	21.37	1.05	1.20	6.86	7.39	4.86	5.05	37.42	33.44
	Overall	HL	1080	15.05	14.33	12.00	7.86	0.32	0.41	2.72	3.28	1.45	1.82	20.81	15.88
		LL	1080	46.50	44.27	42.82	30.88	0.99	1.23	4.98	6.23	3.83	4.50	53.68	42.97
Qwen2.5-VL	2-5 steps	HL	281	22.21	24.35	21.55	20.89	0.29	0.40	1.88	2.06	1.24	1.28	37.15	34.75
		LL	281	51.63	53.12	50.85	52.93	0.63	0.80	2.75	3.05	2.30	2.54	63.86	64.52
	6-8 steps	HL	226	15.10	19.33	12.37	12.49	0.39	0.56	2.68	2.70	1.37	1.39	21.04	21.03
		LL	226	45.06	50.03	43.25	43.41	1.01	1.36	4.55	4.61	3.48	3.53	53.40	53.40
	9-12 steps	HL	222	12.16	13.87	7.90	7.48	0.32	0.38	2.83	2.85	1.26	1.26	15.56	15.17
		LL	222	45.56	49.67	42.14	41.30	1.15	1.43	5.64	5.72	4.39	4.43	53.31	52.69
	13+ steps	HL	351	12.88	13.68	8.30	7.32	0.43	0.49	3.23	3.31	2.02	2.10	15.08	13.54
		LL	351	35.81	37.94	28.97	26.59	1.19	1.37	7.33	8.00	5.59	6.20	39.99	37.52
	Overall	HL	1080	15.62	15.94	12.52	9.17	0.36	0.47	2.68	3.03	1.52	1.78	22.17	16.62
		LL	1080	43.87	44.35	40.36	33.83	1.00	1.32	5.21	6.70	4.04	5.23	51.74	44.78
OS-Atlas-Pro	2-5 steps	HL	281	13.49	13.44	13.11	11.46	0.16	0.19	1.66	1.81	1.08	1.12	32.82	30.08
		LL	281	52.94	54.80	52.35	51.40	0.66	0.82	2.76	3.02	2.45	2.67	69.29	68.51
	6-8 steps	HL	226	8.85	12.19	6.78	6.73	0.23	0.35	1.98	1.99	0.97	0.97	15.51	15.19
		LL	226	41.84	47.73	38.58	38.03	0.95	1.29	4.15	4.18	3.41	3.42	53.73	53.04
	9-12 steps	HL	222	9.61	11.82	6.66	6.50	0.28	0.36	2.41	2.44	1.13	1.13	13.30	13.03
		LL	222	34.86	39.96	29.28	28.42	0.91	1.16	4.47	4.50	3.61	3.63	44.57	43.77
	13+ steps	HL	351	9.21	9.57	5.35	4.80	0.30	0.34	3.20	3.43	1.60	1.74	11.23	10.37
		LL	351	26.14	27.39	18.65	16.55	0.86	0.97	4.95	5.07	4.03	4.19	30.16	27.17
	Overall	HL	1080	10.33	10.96	7.94	5.94	0.25	0.33	2.38	2.91	1.24	1.47	18.17	13.17
		LL	1080	38.19	36.83	33.77	24.57	0.84	1.05	4.11	4.67	3.40	3.85	48.23	37.21
InfGUI	2-5 steps	HL	281	6.45	7.67	5.80	5.49	0.09	0.13	1.19	1.29	0.77	0.78	23.56	21.47
		LL	281	9.46	16.19	8.61	9.61	0.19	0.39	2.05	2.28	1.76	1.92	48.81	49.12
	6-8 steps	HL	226	5.86	9.34	4.04	4.34	0.18	0.30	1.74	1.75	0.70	0.72	10.82	11.01
		LL	226	24.17	34.29	20.19	21.04	0.74	1.18	3.51	3.50	2.87	2.86	43.80	43.30
	9-12 steps	HL	222	8.24	10.37	4.83	4.79	0.23	0.31	2.09	2.08	0.90	0.88	11.11	10.78
		LL	222	26.96	35.52	19.90	20.17	0.83	1.19	3.99	4.01	3.10	3.12	38.16	38.26
	13+ steps	HL	351	6.83	7.60	3.53	3.09	0.24	0.29	2.95	3.25	0.96	0.99	7.38	6.60
		LL	351	24.19	25.71	16.72	14.98	0.82	0.97	4.96	5.11	3.67	3.76	27.85	24.68
	Overall	HL	1080	6.82	8.48	4.50	3.74	0.19	0.28	2.06	2.71	0.84	0.92	13.08	9.03
		LL	1080	20.92	28.24	15.99	16.27	0.64	0.99	3.70	4.50	2.89	3.39	38.76	31.30
InfGUI tk	2-5 steps	HL	281	5.92	8.52	5.28	5.98	0.10	0.20	1.45	1.58	0.96	0.98	28.84	26.76
		LL	281	9.25	17.90	8.23	10.39	0.21	0.47	2.16	2.44	1.93	2.16	52.27	54.02
	6-8 steps	HL	226	8.45	12.93	6.35	6.91	0.25	0.41	2.15	2.16	1.04	1.06	15.97	16.27
		LL	226	27.45	38.96	22.88	23.79	0.83	1.31	4.00	4.01	3.33	3.35	51.12	50.95
	9-12 steps	HL	222	10.75	13.78	6.59	6.75	0.30	0.42	2.37	2.36	1.14	1.14	14.20	14.06
		LL	222	31.04	40.58	23.44	23.76	0.95	1.35	4.70	4.72	3.75	3.77	45.67	45.77
	13+ steps	HL	351	10.11	11.17	5.86	5.14	0.35	0.43	3.54	3.97	1.37	1.36	10.65	9.44
		LL	351	31.10											

E Appendix: Action Type Experiments

Here we provide detailed experimental Results of Agent’s performance across different action types in compensation to Section 3.4 Q2 discussions.

Model	Action Type	Eval Level	Count	Type Acc (%)	Exact Acc (%)	
AGUVIS-7B	CLICK	LL	7542	99.84	91.65	
		HL	7269	67.48	34.68	
	TYPE	LL	1122	95.28	91.89	
		HL	1063	28.69	23.52	
	SCROLL	LL	1563	99.04	87.33	
		HL	1507	14.27	11.02	
	STOP	LL	1460	86.85	86.85	
		HL	1422	0.00	0.00	
	LONG_PRESS	LL	74	94.59	48.65	
		HL	70	1.43	1.43	
	PRESS	LL	262	93.13	93.13	
		HL	355	53.80	53.80	
	Overall		LL	12023	97.55	90.29
			HL	11686	48.07	26.78
	AgentCPM-GUI-7B	CLICK	LL	7545	97.27	88.30
HL			7579	86.50	54.24	
TYPE		LL	1105	89.32	84.98	
		HL	1121	64.67	52.45	
SCROLL		LL	1564	98.27	96.61	
		HL	1561	45.42	41.83	
STOP		LL	1460	79.79	79.79	
		HL	1461	66.19	66.19	
LONG_PRESS		LL	73	82.19	36.99	
		HL	74	2.70	1.35	
PRESS		LL	260	29.62	29.62	
		HL	231	39.39	39.39	
Overall		LL	12007	92.99	86.46	
		HL	12027	75.25	53.31	
UI-TARS-SFT-7B		CLICK	LL	7517	99.16	79.01
	HL		7464	88.45	54.74	
	TYPE	LL	1124	94.40	89.15	
		HL	1121	64.32	53.35	
	SCROLL	LL	1101	99.00	94.19	
		HL	1196	52.34	47.74	
	STOP	LL	1460	99.04	99.04	
		HL	1455	44.81	44.81	
	LONG_PRESS	LL	74	0.00	0.00	
		HL	74	0.00	0.00	
	PRESS	LL	291	53.95	53.95	
		HL	281	35.23	35.23	
	Overall		LL	11567	96.90	82.83
			HL	11591	75.06	51.82

Continued on next page

Table 6 – Continued from previous page

Model	Action Type	Eval Level	Count	Type Acc (%)	Exact Acc (%)
Qwen2.5-VL-7B	CLICK	LL	7562	93.12	83.21
		HL	7564	67.41	44.79
	TYPE	LL	1124	84.34	81.49
		HL	1123	53.25	48.00
	SCROLL	LL	1564	81.01	46.04
		HL	1561	15.25	11.40
	STOP	LL	1460	89.86	89.86
		HL	1465	79.66	79.66
	LONG_PRESS	LL	73	60.27	50.68
		HL	74	0.00	0.00
	PRESS	LL	239	51.46	51.46
		HL	231	25.54	25.54
	Overall	LL	12022	89.30	78.19
		HL	12018	59.59	44.36
OS-Atlas-Pro-7B	CLICK	LL	7364	94.96	74.66
		HL	7587	86.37	48.53
	TYPE	LL	1054	85.67	82.16
		HL	1123	54.50	44.43
	SCROLL	LL	1517	48.65	28.94
		HL	1561	29.85	26.71
	STOP	LL	1459	67.31	67.31
		HL	1457	35.55	35.55
	LONG_PRESS	LL	71	57.75	32.39
		HL	74	0.00	0.00
	PRESS	LL	224	72.32	72.32
		HL	223	57.85	57.85
	Overall	LL	11689	84.00	68.18
		HL	12025	68.84	43.62
InfiGUI-R1-3B tk	Click	LL	7576	97.94	88.79
		HL	7576	88.56	55.16
	Type	LL	1125	89.78	86.84
		HL	1125	57.24	49.33
	Scroll	LL	1561	96.28	7.69
		HL	1561	41.38	23.45
	Complete	LL	1448	8.91	8.91
		HL	1448	9.05	8.98
	Long Press	LL	74	87.84	58.11
		HL	74	1.35	1.35
	Navigate Home	LL	209	90.91	0.96
		HL	209	48.80	42.58
	Navigate Back	LL	22	90.91	86.36
		HL	22	54.55	18.18
	Wait	LL	13	100.00	100.00
		HL	13	7.69	7.69
	Impossible	LL	1	0.00	0.00
		HL	1	0.00	0.00

Continued on next page

Table 6 – Continued from previous page

Model	Action Type	Eval Level	Count	Type Acc (%)	Exact Acc (%)
	Overall	LL	12029	86.04	66.76
		HL	12029	68.55	44.27
InfiGUI-R1-3B	Click	LL	7576	87.04	79.14
		HL	7576	75.17	46.57
	Type	LL	1125	85.16	82.67
		HL	1125	54.22	47.91
	Scroll	LL	1561	81.81	5.64
		HL	1561	32.42	20.24
	Complete	LL	1448	10.43	10.43
		HL	1448	17.47	17.40
	Long Press	LL	74	79.73	47.30
		HL	74	1.35	1.35
	Navigate Home	LL	209	88.52	4.31
		HL	209	45.93	39.71
	Navigate Back	LL	22	77.27	72.73
		HL	22	40.91	18.18
	Wait	LL	13	100.00	100.00
		HL	13	7.69	7.69
	Impossible	LL	1	0.00	0.00
		HL	1	0.00	0.00
	Overall	LL	12029	76.93	60.17
		HL	12029	59.61	39.27

Table 6: Model performance on different action types

F Appendix: Platform Experiments

Here we provide detailed experimental Results of Agent’s performance across different platforms in compensation to Section 3.4 Q2 discussions.

Model	Level	Metric	iOS Devices	Android Devices	Overall
AgentCPM-GUI-8B	LL	EM	75.49%	82.96%	79.32%
		TM	89.11%	91.11%	90.13%
	HL	EM	42.02%	55.19%	48.77%
		TM	68.87%	80.74%	74.95%
UI-TARS-7B	LL	EM	73.54%	86.38%	79.96%
		TM	94.16%	99.22%	96.69%
	HL	EM	39.30%	49.02%	44.14%
		TM	73.15%	78.43%	75.78%
Qwen2.5-VL-7B	LL	EM	66.15%	71.11%	68.69%
		TM	81.32%	85.93%	83.68%
	HL	EM	33.85%	37.04%	35.48%
		TM	53.31%	57.78%	55.60%
OS-Atlas-Pro-7B	LL	EM	63.89%	62.31%	63.08%
		TM	86.51%	82.09%	84.23%
	HL	EM	34.63%	39.63%	37.19%
		TM	64.59%	66.30%	65.46%
AGUVIS-7B	LL	EM	82.10%	88.89%	85.58%
		TM	95.72%	99.26%	97.53%
	HL	EM	19.43%	27.17%	23.44%
		TM	53.44%	57.36%	55.47%
InfiGUI-R1-3B thinking	LL	EM	66.15%	65.70%	65.92%
		TM	81.54%	90.97%	86.41%
	HL	EM	32.69%	37.91%	35.38%
		TM	65.77%	67.87%	66.85%
InfiGUI-R1-3B	LL	EM	59.23%	54.87%	56.98%
		TM	70.00%	78.70%	74.49%
	HL	EM	29.62%	38.99%	34.45%
		TM	56.15%	64.98%	60.71%

Table 7: Model performance on iOS and Android devices

G Appendix: Language Experiments

Here we provide detailed experimental Results of Agent’s performance across different languages in compensation to Section 3.4 Q2 discussions.

Model Name	Language	Level	TM	EM	EN-CN Difference
OS-Atlas-Pro-7B	CN	HL	65.09	40.88	TM: +0.36, EM: -0.43
	EN	HL	65.45	40.45	
	CN	LL	83.68	66.86	TM: 0.00, EM: +2.93
	EN	LL	83.68	69.79	
Qwen2.5-VL-7B	CN	HL	64.53	45.76	TM: -5.96, EM: -2.46
	EN	HL	58.57	43.30	
	CN	LL	95.11	86.14	TM: -3.09, EM: -4.54
	EN	LL	92.02	81.60	
AGUVIS-7B	CN	HL	47.95	26.65	TM: -4.36, EM: -1.61
	EN	HL	43.59	25.04	
	CN	LL	96.69	88.93	TM: +0.87, EM: +2.95
	EN	LL	97.56	91.88	
UI-TARS-7B	CN	HL	70.61	45.57	TM: -1.24, EM: +3.01
	EN	HL	69.37	48.58	
	CN	LL	96.76	80.44	TM: -0.21, EM: +5.67
	EN	LL	96.55	86.11	
AgentCPM-GUI-8B	CN	HL	72.63	50.07	TM: +0.14, EM: +3.02
	EN	HL	72.77	53.09	
	CN	LL	93.24	88.21	TM: +0.15, EM: +0.29
	EN	LL	93.39	88.50	
InfGUI-R1-3B	CN	HL	48.64	30.96	TM: +7.39, EM: +6.76
	EN	HL	56.03	37.72	
	CN	LL	72.84	55.17	TM: +2.16, EM: +3.38
	EN	LL	75.00	58.55	
InfGUI-R1-3B thinking	CN	HL	65.37	41.31	TM: -2.01, EM: -0.58
	EN	HL	63.36	40.73	
	CN	LL	87.28	65.73	TM: -3.95, EM: -1.87
	EN	LL	83.33	63.86	

Table 8: Model performance on Chinese(CN) and English(EN) instructions