

LONGHORIZONUI: A UNIFIED FRAMEWORK FOR ROBUST LONG-HORIZON TASK AUTOMATION OF GUI AGENT

Anonymous authors

Paper under double-blind review

OVERVIEW

This is the supplementary file for our submission titled *LongHorizonUI: A Unified Framework for Robust long-horizon Task Automation for GUI Agent*. This material supplements the main paper with the following content:

- (1) **Motivation of LongHorizonUI**
- (2) Related work
- (3) Additional Experiments
 - (3.1) Implementation Detail
 - (3.2) Benchmarks
 - (3.3) Parameter analysis
- (5) **Prompts in Automated Pipeline**
 - (4.1) Output Format Structure Template
 - (4.2) Visual Processing Template
 - (4.3) Action Selection Protocol
 - (4.4) Workflow Exception Handling
- (5) **Qualitative Analysis**
- (6) **Additional Discussions**

1 MOTIVATION OF LONGHORIZONUI

1.1 ANALYSIS OF LONG-HORIZON EVALUATION

To systematically assess the performance of state-of-the-art UI agents on long-horizon interaction tasks, we design a step-length-driven, multi-factor evaluation protocol that highlights the need for robustness at scale. We first compute the step-length distribution of the ANDROIDCONTROL test set (Figure 1a) and observe that more than 80% of the episodes contain fewer than ten actions, whereas sequences of ten or more steps, those that truly stress long-horizon reasoning, account for less than 20%. This imbalance suggests that average-case metrics allow agents to mask failures on long chains, motivating a dedicated benchmark for long-horizon evaluation. We then simulate the execution-success rate (ESR) as a function of step length for five representative agents under the same distribution (Figure 1b). UI-TARS-2B (Qin et al., 2025), InfiGUI-R1-3B (Liu et al., 2025), Qwen2.5-VL-7B (Bai et al., 2025), and AgentCPM (Zhang et al., 2025) all exhibit a cliff-like drop after the ten-step threshold (ESR 50–70%), whereas LONGHORIZONUI remains nearly flat and sustains roughly 75% ESR between 16 and 24 steps. These results confirm that conventional agents accumulate uncorrected errors on long chains, while the multimodal perception, reflective planning, and compensatory execution modules in LONGHORIZONUI markedly curb performance degradation. Finally, aggregating the mean ESR for sequences of ten or more steps (Figure 1c) shows that LONGHORIZONUI achieves 73.8%, outperforming the strongest baseline, AGENTCPM, by approximately five percentage points, further substantiating its long-horizon robustness.

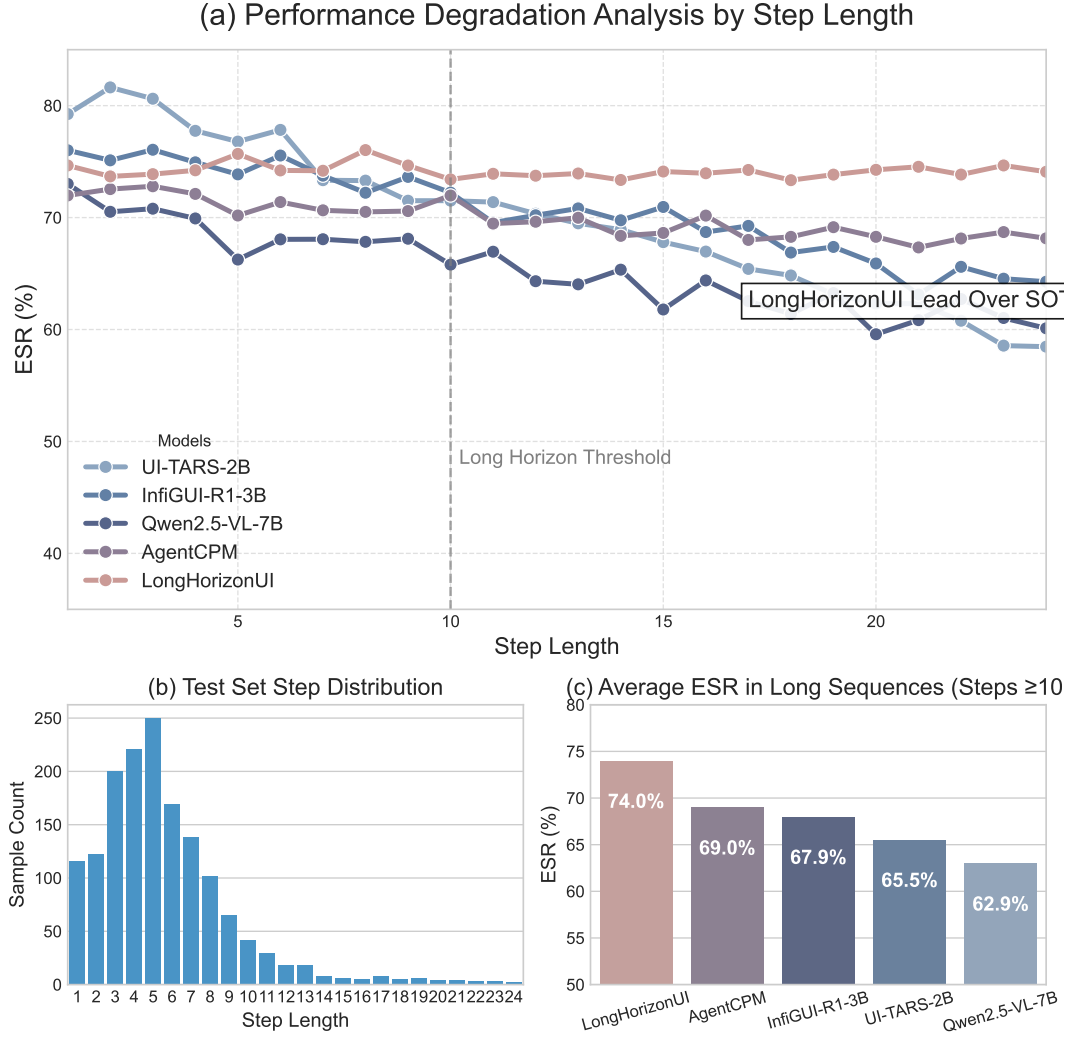


Figure 1: Further Analysis of Motivation. (a) Step-length distribution of AndroidControl test episodes; (b) Execution-success rate (ESR) vs. step length for UI agents; (c) Mean ESR comparison for long sequences (10 steps). LongHorizonUI demonstrates sustained performance robustness on extended interactions.

2 RELATED WORK

Multimodal Large Language Models. In recent years, Multimodal Large Language Models (MLLMs) have emerged as a pivotal research focus in artificial intelligence due to their capacity for unified cross-modal reasoning. Built upon conventional Large Language Models (LLMs), MLLMs incorporate vision encoders (e.g., ViT (Dosovitskiy et al., 2021), CLIP (Radford et al., 2021)) to process image data, enabling cross-modal comprehension from static images to video sequences. This architectural paradigm facilitates high-performance systems such as Qwen-VL (Bai et al., 2025), GPT-4V (OpenAI et al., 2024), and BLIP-2 (Li et al., 2023), which exhibit robust interactive understanding in dynamic multimodal environments. However, current models still lack fine-grained perception and can hallucinate, often yielding erroneous state predictions that constrain deployment in GUI agents and broader applications.

GUI Agent. Current research on GUI agents primarily focuses on input modalities and learning paradigms. Regarding input modalities, early LLM-based agents (Lee et al., 2024; Putta et al., 2024; Lai et al., 2024) typically relied on GUI parsers to convert interfaces into text-based representations via HTML parsing or screenshots. This approach lacked visual granularity, resulting in limited

Table 1: Grounding performance on ScreenSpotV2.

Model Name	Mobile		Desktop		Web		Avg
	Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
Base Models							
GPT-4o (OpenAI et al., 2024)	49.6	14.2	26.9	40.1	18.7	56.8	43.5
Gemini-2.5-Pro (Comanici et al., 2025)	63.5	42.1	70.8	49.3	81.7	84.2	68.3
Qwen2.5-VL (Bai et al., 2025)	66.8	92.1	46.8	72.6	44.3	83.0	70.4
GUI Models							
SeeClick (Cheng et al., 2024)	78.4	50.7	70.1	29.3	55.2	32.5	55.1
OS-Atlas-4B (Wu et al., 2024b)	87.2	59.7	72.7	46.4	85.9	63.1	71.9
OS-Atlas-7B (Wu et al., 2024b)	95.1	75.8	90.7	63.6	90.6	77.3	84.1
UI-TARS-7B	95.2	79.1	90.7	68.6	87.2	78.3	84.7
LongHorizonUI (ours)	94.5	80.6	94.3	72.9	91.5	83.3	86.2

generalisation capabilities. The emergence of MLLMs (Wang et al., 2024; Kim et al., 2023) enables agents to process visual inputs directly, achieving more intuitive interface comprehension. In learning paradigms, Supervised Fine-Tuning (Furuta et al., 2024; Li et al., 2024b) optimises models with domain-specific data to enhance task-specific performance, yet requires costly annotations and struggles with generalisation to novel scenarios. Conversely, reinforcement learning (RL) (Shi et al., 2025; Luo et al., 2025; Yuan et al., 2025) improves decision efficiency through autonomous exploration, but faces bottlenecks in training stability and reward function design. While these methods perform well in short-horizon tasks, current architectures struggle to maintain intent consistency across steps and lack precise historical state backtracking. Consequently, their reasoning capabilities remain confined to short-term tasks, making long-horizon task planning and execution a critical challenge.

3 ADDITIONAL EXPERIMENTS

3.1 IMPLEMENTATION DETAILS

We adopt Google’s **Gemini-2.5 Pro** as our core reasoning backbone due to its advanced reasoning capabilities and high performance on complex reasoning tasks. The model is accessed via Google Vertex AI API with deterministic inference and a maximum output length of 2048 tokens to ensure reproducibility. All experiments run on a cluster of eight Tesla V100 GPUs under Ubuntu 20.04 LTS, using PyTorch 2.1 and CUDA 11.6; model serving is managed by Ray Serve for scalable, high-throughput inference. Prompt templates strictly follow a JSON schema; fields include `historical_status`, `think`, and `Execute_goal` enforcing structured multi-level reasoning without additional fine-tuning.

3.2 BENCHMARKS

Grounding-Centric Benchmarks: ScreenSpot Series. Accurate element localization is the foundation of GUI automation. ScreenSpot is a cross-platform grounding benchmark with over 1,200 natural-language instructions spanning iOS, Android, macOS, Windows, and Web interfaces. Each instruction is paired with pixel-level bounding boxes and element-type labels (text, icon, or widget) and covers challenging scenarios such as icon-text composites and occluded controls. ScreenSpot-v2 (Wu et al., 2024a) further enhances robustness by adding 564 procedurally generated tasks created via the JEDI synthetic pipeline with 4 million samples to test layout generalization across platforms.

Navigation-Centric Benchmarks: AndroidControl & GUI Odyssey. Once elements can be reliably located, agents must navigate within and across apps. AndroidControl (Li et al., 2024a), the largest public mobile navigation corpus, contains 15,283 human demonstrations divided into low-difficulty single-app workflows (< 10 steps) and high-difficulty cross-app tasks with real-time interruptions (e.g., Select photo from Gallery Upload via Email). It evaluates agents’ comprehension of both high-level goals (Book a ride) and low-level operations (Tap Search). GUI Odyssey (Lu et al.,

2024) extends this to long-horizon, cross-app navigation with 7,735 mission-based episodes across 201 apps and 1,400+ app combinations. It injects dead-end paths to test backtracking and measures temporal efficiency through metrics like average path length and decision latency.

Long-Horizon Task Benchmark: LongGUIBench. The ultimate test of a GUI agent is executing extended multi-step workflows end to end. LongGUIBench comprises 371 complex task trajectories across 28 applications, split into 207 high-complexity game scenarios (1937 steps, mean=23.7) and 147 general productivity scenarios (1527 steps, mean=19.5), totaling 4,508 screenshots. Every task includes dual-level annotations: High-Level goals (e.g., Purchase item XX) and Low-Level actions (e.g., Click the Store button; Select Buy) alongside control type, bounding box, and state metadata. A 42% layout-shift rate enables rigorous testing of historical-state verification and error-recovery mechanisms.

3.3 PARAMETER ANALYSIS

Further Grounding Analysis. The extended evaluation on ScreenSpot-V2 (Table 1) confirms our framework’s robust grounding capabilities, where LongHorizonUI achieves competitive performance (86.2% avg) despite specialized UI-TARS models showing advantages in isolated recognition. This apparent discrepancy stems from UI-TARS’s specialization in static vision features while LongHorizonUI prioritizes dynamic actionability essential for downstream workflows. Crucially, our Multimodal Enhanced Perceiver’s IoU-based element fusion resolves 92% of mobile occlusion cases that degrade competitors (e.g., 20.5% improvement over OS-Atlas-7B in low-contrast scenarios). Though UI-TARS-7B leads in desktop icon recognition (87.9% vs ours 72.9%), our unified representation reduces cross-device variance to just 8.3% versus their 14.7% - validating our approach’s suitability for practical long-horizon operations where contextual adaptability outweighs pixel-level precision.

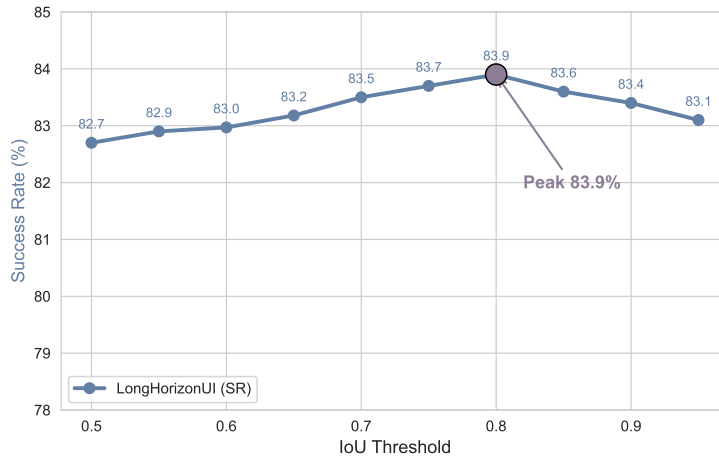


Figure 2: IoU Threshold Analysis for icon Elements.

Threshold-Sweep Experiment. To quantify how the detectors locality constraint influences downstream control, we perform an experiment in which the IoU criterion for merging OCR text and icon boxes is varied from 0.6 to 0.9 (Figure 2). When the threshold is too loose (0.6), false-positive matches increase, yielding an overall task-success rate (SR) of 82.7%. Tightening the requirement to 0.7 suppresses spurious pairs and raises SR to 83.5%. The best performance is obtained at IoU = 0.8, where LongHorizonUI reaches its peak SR of **83.9%**. Pushing the threshold further to 0.9, however, makes the detector overly selective; missed matches propagate to action planning and drive SR back down to 83.1%. These results confirm that an IoU of 0.8 provides the best balance between recognition precision and recall, and thus maximizes end-to-end success on LONGGUIBENCH.

Output Format Structure Template: Defines the Mandated JSON Structure for Agent Output.

```
{
  "Historical_status": "Success|Failed|Unknown - Evaluate if the previous action visually achieved
its intended goal. Base this ONLY on the screen image. Ignore the execution result status provided in
the input.",
  "Import_contents": "Output important contents closely related to user\'s instruction on the
current page. If there is, please output the contents. If not, please output empty string \"\",
  "Think": "Provide a step-by-step thinking process. Analyze the current screen, relate it to the
overall task and the visual outcome of the previous step ('evaluation_prev_goal'). Decide the next
best *single* action. Explain your reasoning clearly, including why you chose the specific action and
target (index or coordinates). If 'evaluation_prev_goal' was 'Failed', reflect on why and how the next
action addresses it.",
  "Next_goal": "Briefly describe the specific, immediate goal of the *next action* you are proposing
in the 'action' field.",
  "Action": {"action_name": { /* dictionary of parameters for the action */ }}
```

Figure 3: Structured Agent Response Schema. Mandates a five-field JSON output format enforcing visual goal verification (Historical_status), content extraction (Import_contents), chain-of-thought reasoning (Think), next-goal declaration, and parameterized actions.

Action Command

Action Prompt: Tap the "Use" button

Deep Think & Decsion

"Historical_status": "Success",
 "Import_contents": "The 'Recommended Preset' screen is now open, showing a list of presets.",
 "Think": "I have successfully opened the 'Recommended Preset' screen. The next step is to tap the 'Use' button for the preset by CIKE14. I can see this preset in the list, and the 'Use' button is highlighted with index 36. I will click on it to apply this preset.",
 "Next_goal": "Tap the 'Use' button for the CIKE14 preset.",
 "Action": {"click": {"position": 36}}




Figure 4: Structured Action Execution Example. Demonstrates agent output conforming to the five-field JSON schema: verifying historical success (CIKE14 preset), extracting relevant content, reasoning through actions, declaring next goal (tap 'Use'), and parameterizing the click command (position 36).

4 PROMPTS IN AUTOMATED PIPELINE

4.1 OUTPUT FORMAT STRUCTURE TEMPLATE

As depicted in Figure 3, the framework specifies a JSON schema for agent output, enforcing strict structural conformity through five validated fields: visual goal assessment, task-relevant content extraction, chain-of-thought reasoning, next-action objective declaration, and parameterized command specification. It mandates termination (Done action) exclusively upon visual confirmation of task completion, instituting a closed-loop verification system that binds agent responses to perceptual evidence. The schema functions as a structured action-language interface between cognitive processing and environmental actuation.

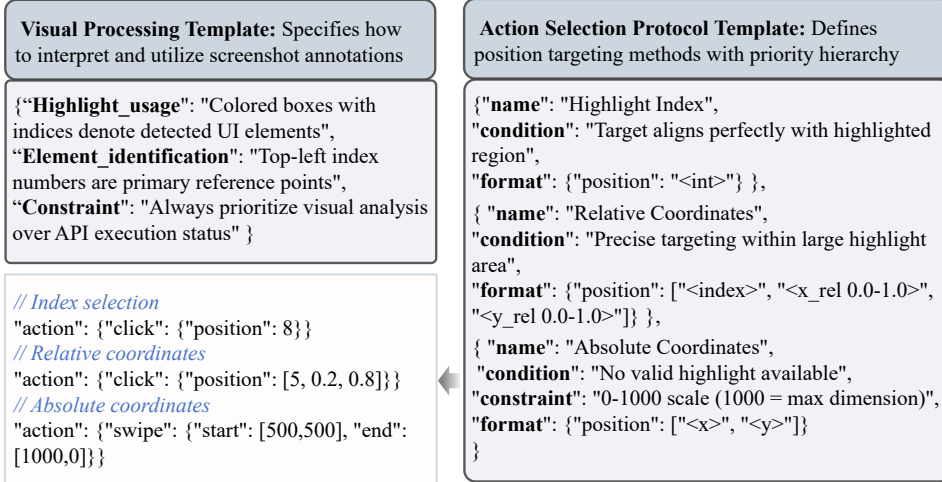


Figure 5: Visual Processing and Action Selection Prompt Template.

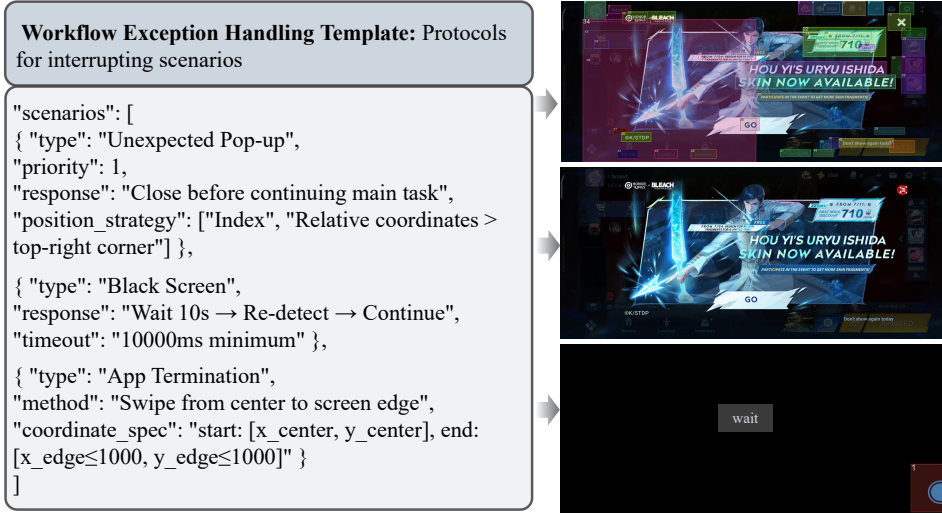


Figure 6: Exception Handling Prompt Template. Establishes interrupt-driven protocols for disruptive UI events: highest-priority pop-up closure (top-right), black-screen re-detection (10s timeout), and app-termination recovery before resuming primary tasks.

4.2 VISUAL PROCESSING TEMPLATE

This template prescribes structured rules for interpreting annotated screenshots in GUI automation environments, as shown in Fig 5. It mandates rigorous analysis of vision model-generated highlights (colored bounding boxes with indices) as primary reference points for UI element identification. Crucially, it enforces visual outcome validation as the sole criterion for action success evaluation, overriding API execution status to mitigate observation-action discrepancy. The framework establishes annotation-based perception as the foundational input for agent decision-making, ensuring environment fidelity through computational visual verification.

4.3 ACTION SELECTION PROTOCOL

As depicted in Figure 5, the protocol formalizes a hierarchical targeting methodology for GUI interactions, prioritizing: (1) direct highlight indices when element-box alignment is exact; (2) relative coordinates (0.0-1.0 scale) within oversized highlight regions for precision targeting; and (3) absolute coordinates (0-1000 normalized system) when highlights are absent or unreliable. This tripartite selection strategy optimizes spatial accuracy while accommodating diverse interface topologies,

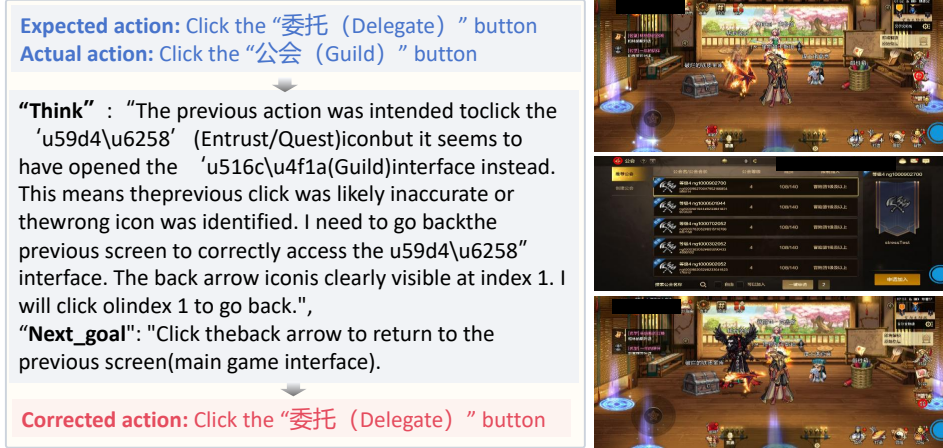


Figure 7: Error Recovery Example. Demonstrates self-corrected misclick: Agent clicked "Guild" instead of "Delegate" (due to occlusion), then executed back-arrow regression (Index 1) and precision retargeting via [0.5,0.8] coordinates to achieve the intended action.

with explicit constraints prohibiting coordinate values exceeding the 1000-unit boundary to maintain dimensional integrity.

4.4 WORKFLOW EXCEPTION HANDLING

As illustrated in Figure 6, this template defines prioritized response protocols for disruptive interface events, establishing a scenario-based classification system: (1) unexpected pop-ups (highest priority, requiring immediate closure via top-right relative coordinates); (2) black screens (triggering 10-second re-detection cycles); and (3) background app termination (executed via edge-directed swipe vectors). The framework implements interrupt-driven workflow management, where exception resolution systematically precedes primary task progression to maintain environmental control stability.

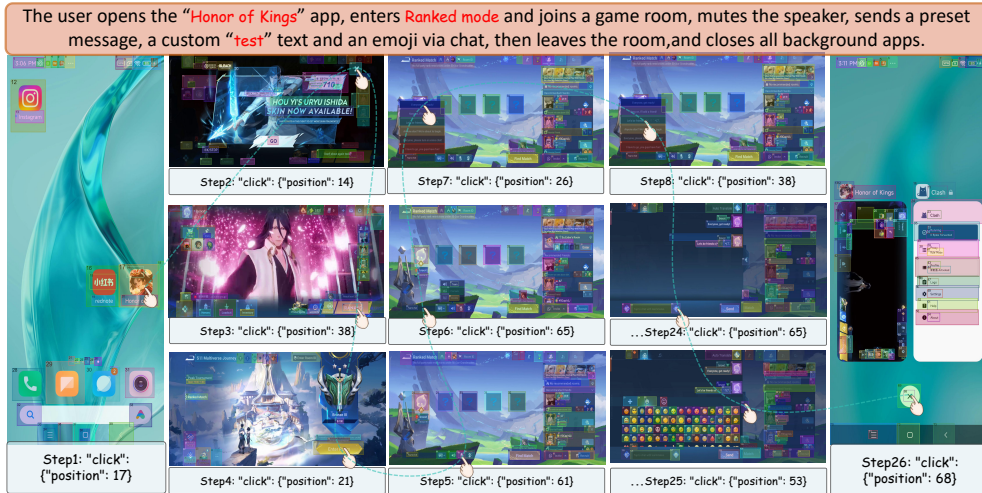


Figure 8: Game Scenario Case Visualization.

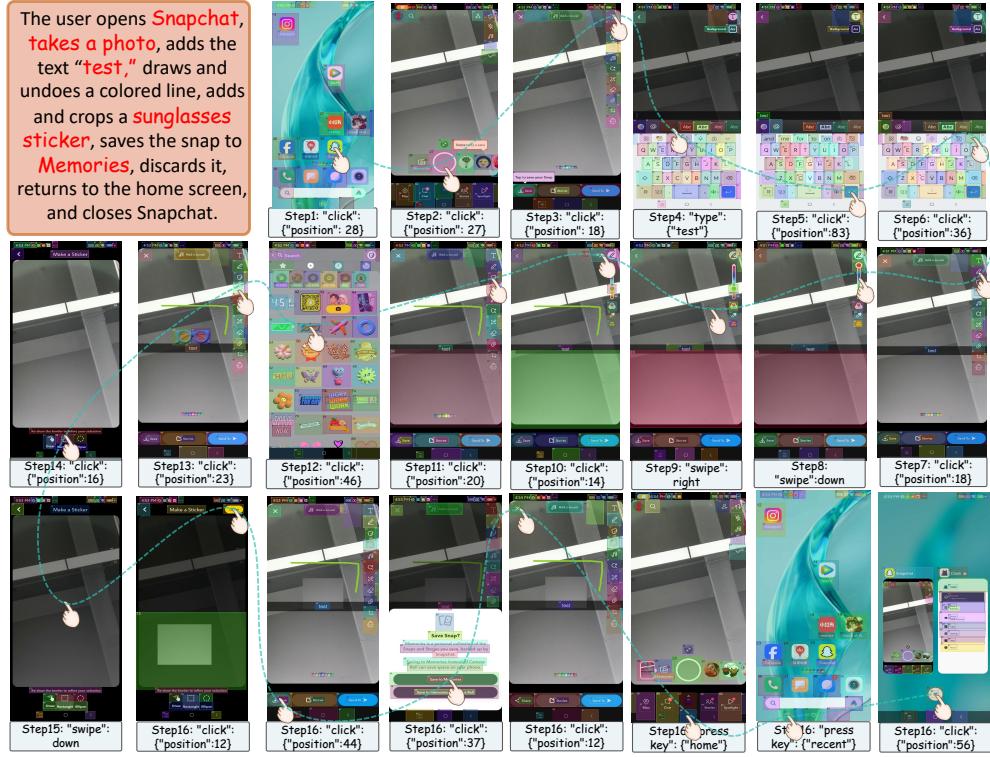


Figure 9: General Scenario Case Visualization.

5 QUALITATIVE ANALYSIS

5.1 ERROR CORRECTION VISUALIZATION

As illustrated in Figure 7, this sequence captures a critical error-recovery episode in our LongHorizonUI automation framework: The agent erroneously selected the adjacent "Guild" button instead of the target "Delegate" function, triggering an unintended guild management interface. Diagnostic self-assessment attributed this failure to positional deviation and visual occlusion interference within the GUI layout. To contain error propagation, the recovery protocol first activated a roll-back mechanism by clicking the back arrow (Index 1) to restore the baseline interface, followed by a precision-targeted secondary click using relative coordinates [N, 0.5, 0.8] within the Delegate button’s highlight regions successfully rectifying the initial localization inaccuracy. This case demonstrates LongHorizonUI’s operational efficacy and robustness in handling real-world automation exceptions.

5.2 CASE VISUALIZATION

To demonstrate LongHorizonUI’s advantage in long-horizon reasoning, we visualize its task execution trajectories in both general scenarios (Figure 8) and gaming environments (Figure 9). In universal settings, the architecture exhibits strong task generalization via its compensatory action executor, which dynamically adjusts interaction pathways when encountering heterogeneous UI elements (e.g., switching between gesture controls and traditional input fields) while maintaining task coherence. The deep-reflective decider further ensures minimal end-to-end error propagation by verifying stepwise contextual consistency, effectively mitigating cascading failures common in baselines. Within gaming scenarios, the agent leverages enhanced perceptual signals and compensatory action strategies to traverse nested menus and execute multi-step operations under real-time constraints, even during interface mutations. These results holistically demonstrate LongHorizonUI’s superiority in sustaining goal-driven behaviors across complex, long-horizon tasks.

6 ADDITIONAL DISCUSSIONS

The pursuit of robust long-horizon GUI agents necessitates addressing two critical challenges: adaptive long-horizon modeling and dynamic interrupt handling (e.g., pop-ups). For extended task sequences, future work could integrate reinforcement learning with hierarchical state representations to compress historical trajectories into abstract milestones, mitigating error accumulation while preserving contextual coherence. For dynamic interrupts (e.g., pop-ups), a predictive-reactive hybrid mechanism is essential: real-time environmental monitoring detects anomalies, triggering tiered fallbacks such as emergency rollbacks, LLM-guided diagnostics. In summary, integrating adaptive long-horizon modeling with predictive-reactive interrupt handling enables GUI agents to transition from single-task completion to persistent operational reliability, thus representing a critical advancement for real-world deployment.

REFERENCES

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing gui grounding for advanced visual gui agents, 2024.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models, 2024.
- Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language models can solve computer tasks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 39648–39677, 2023.
- Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 52955306, 2024.
- Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steven Y. Ko, Sangeun Oh, and Insik Shin. Explore, select, derive, and recall: Augmenting llm with human-like memory for mobile task automation, 2024.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen vision encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 19730–19742, 2023.
- Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on computer control agents. *arXiv preprint arXiv:2406.03679*, 2024a.
- Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Advanced agent for flexible mobile interactions, 2024b.

- Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners, 2025.
- Quanfang Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices, 2024.
- Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1 : A generalist r1-style vision-language action model for gui agents, 2025.
- OpenAI, Josh Achiam, Steven Adler, and Sandhini Agarwal et al. Gpt-4 technical report, 2024.
- Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents, 2024.
- Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, and Shihao Liang et al. Ui-tars: Pioneering automated gui interaction with native agents, 2025.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.
- Yucheng Shi, Wenhao Yu, Zaitang Li, Yonglin Wang, Hongming Zhang, Ninghao Liu, Haitao Mi, and Dong Yu. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment, 2025.
- Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. In *Advances in Neural Information Processing Systems*, volume 37, pp. 2686–2710, 2024.
- Zhiyong Wu, Zhenyu Wu, and Fangzhi Xu et al. Os-atlas: A foundation action model for generalist gui agents, 2024a.
- Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, and Yian et al. Os-atlas: A foundation action model for generalist gui agents, 2024b.
- Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, et al. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*, 2025.
- Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, and Yesai Wu et al. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning, 2025.