SOLAR-RL: Semi-Online Long-horizon Assignment Reinforcement Learning

Jichao Wang[125*] **Liuyang Bian**[1*] **Yufeng Zhou**[14*] **Han Xiao**[13]
Yue Pan[1] **Guozhi Wang**[1] **Hao Wang**[15] **Zhaoxiong Wang**[1]
Yafei Wen[1] **Xiaoxin Chen**[1] **Shuai Ren**[1†] **Lingfang Zeng**[2‡]
[1]vivo AI Lab [2]Zhejiang Lab [3]CUHK MMLab [4]Hubei University
[5]Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences
{wangjichao23}@mails.ucas.ac.cn

Abstract

As Multimodal Large Language Models (MLLMs) mature, GUI agents are evolving from static interactions to complex navigation. While Reinforcement Learning (RL) has emerged as a promising paradigm for training MLLM agents on dynamic GUI tasks, its effective application faces a dilemma. Standard Offline RL often relies on static step-level data, neglecting global trajectory semantics such as task completion and execution quality. Conversely, Online RL captures the long-term dynamics but suffers from high interaction costs and potential environmental instability. To bridge this gap, we propose SOLAR-RL (**S**emi-**O**nline **L**ong-horizon **A**ssignment **RL**). Instead of relying solely on expensive online interactions, our framework integrates global trajectory insights directly into the offline learning process. Specifically, we reconstruct diverse rollout candidates from static data, detect the first failure point using per-step validity signals, and retroactively assign dense step-level rewards with target-aligned shaping to reflect trajectory-level execution quality—effectively simulating online feedback without interaction costs. Extensive experiments demonstrate that SOLAR-RL significantly improves long-horizon task completion rates and robustness compared to strong baselines, offering a sample-efficient solution for autonomous GUI navigation.

1 Introduction

The development of autonomous agents capable of mastering Graphical User Interfaces (GUIs) is a pivotal frontier in multimodal AI. Unlike traditional agents confined to specific APIs or DOM-tree parsing, modern GUI agents powered by Multimodal Large Language Models (MLLMs) operate directly on pixel-level visual inputs, promising universal applicability across diverse operating systems (Qin et al., 2025). While agents like CogAgent (Hong et al., 2024) have demonstrated proficiency in perceiving complex interfaces and executing atomic actions, a significant gap remains between "perceiving" an interface and "completing" long-horizon, multi-step workflows in real-world environments.

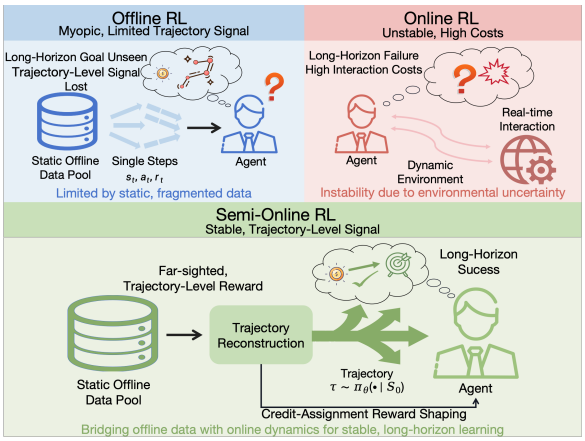


Figure 1: Comparison of RL paradigms for GUI agents. **(Top Left)** Standard Offline RL is limited by fragmented step-level data, leading to temporal myopia and loss of global context. **(Top Right)** Online RL captures dynamics but suffers from instability and prohibitive interaction costs. **(Bottom)** Our SOLAR-RL bridges this gap by retrofitting global trajectory insights into offline data. It utilizes trajectory reconstruction and retroactive credit assignment via failure-point detection, combined with target-aligned reward shaping, to simulate pseudo-online feedback, ensuring stable long-horizon optimization.

Current state-of-the-art (SOTA) approaches primarily rely on Supervised Fine-Tuning (SFT) or Behavior Cloning (BC) from expert demonstrations (Rawles et al., 2023; Li et al., 2024). While SFT effectively initializes agents with basic semantic understanding, it suffers heavily from covariate shift (Ross et al., 2011). When an agent encounters states deviating slightly from the training distribution—a frequent occurrence in dynamic GUIs—it

* Equal contribution
† Project Lead
‡ Corresponding authors

Findings of the Association for Computational Linguistics: ACL 2026, pages 24419–24432

lacks recovery mechanisms, leading to compounding errors. Consequently, recent works have pivoted towards Reinforcement Learning (RL) to endow agents with planning and exploration capabilities (Bai et al., 2024; Wang et al., 2025).

However, as illustrated in Figure 1, applying RL to GUI automation presents a structural dilemma, particularly for long-horizon tasks. Online RL, while theoretically capable of capturing dynamic environmental feedback, suffers from severe instability and high variance when scaling to long trajectories. The prohibitive interaction costs and sparse reward signals in 30+ step workflows often lead to optimization failure before a successful policy can be learned (Bai et al., 2024). Conversely, Offline RL circumvents interaction risks but is plagued by *temporal myopia*. By constraining learning to fragmented, step-level transitions in a static dataset, standard offline methods lose the global context required for long-term planning, making them susceptible to compounding errors (Levine et al., 2020). Critically, both paradigms grapple with the Credit Assignment Problem (CAP) (Lu et al., 2025b). In long-horizon GUI navigation, a sparse binary "success/failure" signal at the trajectory's end is insufficient to assign credit to intermediate reasoning steps. Consequently, gradients vanish or become noisy, leaving agents unable to distinguish critical decisions from irrelevant actions.

This challenge motivates the need for a paradigm that retains the stability of offline learning while incorporating trajectory-level signals typically available only in online interaction. The Semi-Online RL paradigm illustrated in Figure 1 (Right) offers a compromise. Our proposed SOLAR-RL is a concrete instantiation of this paradigm.

SOLAR-RL (**S**emi-**O**nline **L**ong-horizon **A**ssignment **RL**) synthesizes pseudo-online feedback from offline data to address the long-horizon credit-assignment problem. Specifically, we reconstruct diverse rollout candidates from static data and evaluate per-step validity to detect the first failure point where execution breaks down. We then perform retroactive credit assignment by granting positive rewards only to the valid prefix before the failure point while penalizing invalid steps after breakdown. Finally, we apply target-aligned reward shaping to align the total shaped return with trajectory-level execution quality, producing dense and stable training signals without any environment interaction.

Our main contributions are summarized as follows:

- We propose SOLAR-RL, a framework that bridges the gap between offline training stability and online exploration by simulating dynamic feedback mechanisms within static datasets.
- We introduce a trajectory-aware reward shaping mechanism that addresses the credit assignment problem by performing retroactive credit assignment via failure-point detection and target-aligned reward shaping, distilling trajectory-level execution quality into dense step-level rewards.
- We achieve competitive performance with online/SFT baselines without interaction, specifically demonstrating superior robustness and generalization in complex, long-horizon tasks compared to strong baselines.

2 Related Work

2.1 Vision-based GUI Agents

The evolution of GUI agents has shifted from dependency on structured metadata (e.g., DOM trees, View Hierarchies) to purely vision-based approaches that leverage MLLMs. Early works like CogAgent (Hong et al., 2024) introduced high-resolution visual encoders to handle small GUI elements. Recently, the field has moved towards generalist agents capable of cross-platform operation. Qin et al. (2025) proposed UI-TARS, an end-to-end native GUI agent that achieves SOTA performance on benchmarks like OSWorld (Xie et al., 2024) by unifying perception and action spaces.

However, while these models excel at atomic grounding, they often struggle with tasks requiring long-term planning and error correction. A promising trend is the transition from reactive actors to deliberative reasoners. InfiGUI-R1 (Liu et al., 2025) incorporates a "System 2" reasoning stage before action execution, significantly reducing hallucination in complex workflows. Unlike these methods that primarily focus on SFT, our work explores how to further optimize these vision-centric agents through reinforcement learning to handle long-horizon dependencies.

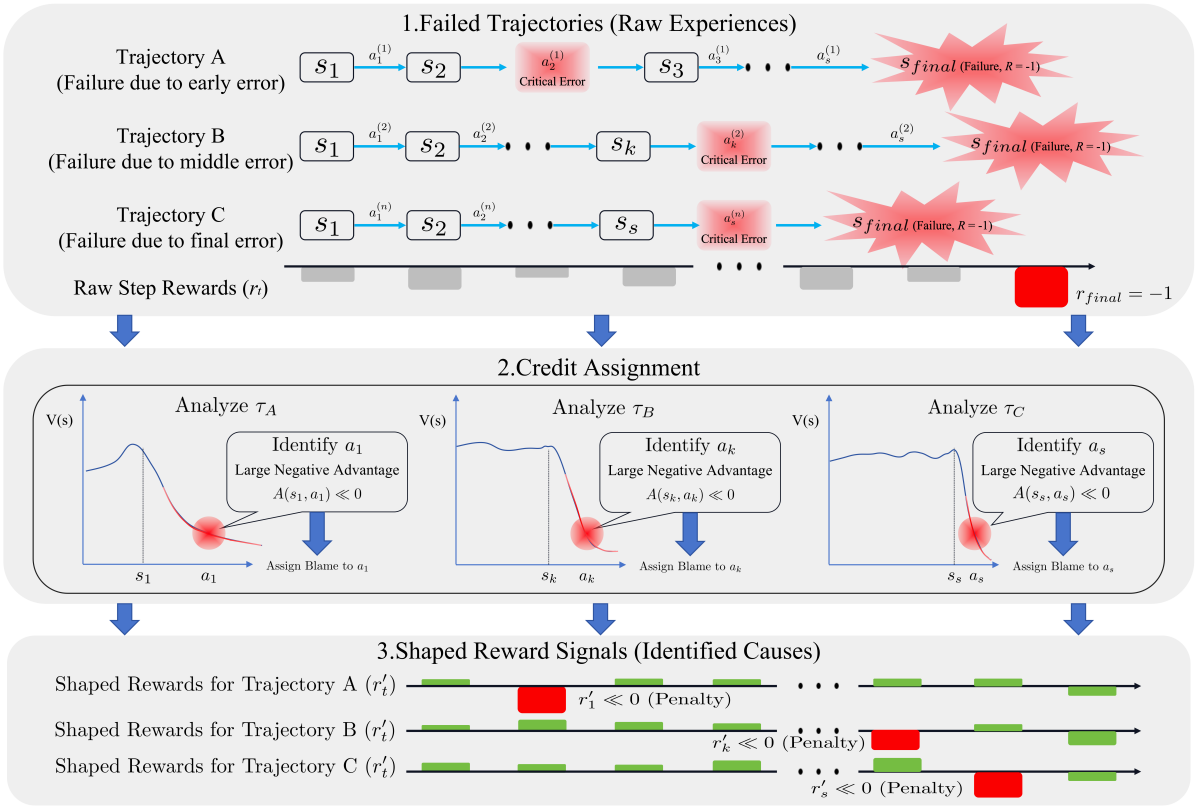


Figure 2: Illustration of the Trajectory-Aware Reward Shaping mechanism. The process consists of three stages: (1) **Raw Experiences:** Failed trajectories usually provide only sparse terminal feedback, obscuring where the execution first goes wrong. (2) **Failure-point Detection:** We identify the first breakdown step where the rollout deviates from valid execution according to per-step validity scores. (3) **Prefix Credit Assignment:** We assign positive rewards only to valid steps *before* the breakdown, while penalizing all invalid steps along the trajectory, producing dense and stable training signals for long-horizon optimization.

2.2 Reinforcement Learning for GUI Automation

While SFT provides a solid initialization, it struggles to generalize to unseen states due to the discrepancy between the expert's policy and the agent's learned policy during inference. RL has been adopted to mitigate this. DigiRL (Bai et al., 2024) pioneered an offline-to-online curriculum but faces high interaction costs in real-time environments. To address the efficiency bottleneck, Semi-Online RL has emerged as a promising paradigm. UI-S1 (Lu et al., 2025b) introduced a framework that simulates online rollouts using static offline trajectories via a "Patch Module" to correct deviations, balancing stability with exploration. MobileGUI-RL (Shi et al., 2025) extends this by optimizing trajectories in decentralized online environments. More recently, UI-Mem (Xiao et al., 2026) augments online GUI RL with a hierarchical experience memory that stores reusable workflows, subtask skills, and failure patterns, improving cross-task and cross-application transfer. Complementing these interaction-efficient paradigms, UI-Genie (Xiao et al., 2025) employs a specialized image-text reward model (UI-Genie-RM) to provide fine-grained, step-level supervision, enabling the iterative generation of high-quality synthetic trajectories.

Most recent approaches employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to eliminate the need for a separate value network, reducing memory overhead. However, standard GRPO often relies on sparse trajectory-level rewards, which we argue are insufficient for long-horizon GUI tasks. Our method builds upon the semi-online paradigm but introduces a more granular, outcome-aware credit assignment mechanism.

2.3 Credit Assignment in Long-Horizon Tasks

The Credit Assignment Problem (CAP) remains the central challenge in RL-based GUI agents, as supervisory signals are often delayed and non-

informative until the episode terminates. Recent works have attempted to densify rewards through Process Reward Models (PRMs).Beyond reward shaping, recent work also explores densifying supervision through training-only distillation. Skill-SD (Wang et al., 2026) summarizes successful trajectories into compact natural-language skills and uses them as dynamic privileged information for teacher-guided token-level supervision, improving stability in multi-turn agent RL. Compared with such distillation-based dense supervision, SOLAR-RL focuses on trajectory-aware reward shaping and retroactive credit assignment directly on semi-online GUI trajectories.

Most notably, VAGEN (Wang et al., 2025) formulates visual agent tasks as Partially Observable Markov Decision Processes (POMDPs) and proposes Bi-Level General Advantage Estimation (Bi-Level GAE). VAGEN explicitly rewards "World Modeling" reasoning (state estimation and transition prediction) and propagates credit at both the turn-level and token-level. Similarly, M-GRPO (Hong et al., 2025a) introduces hierarchical credit assignment for multi-agent systems.

Different from VAGEN's reliance on explicit internal world modeling or the heavy annotation cost of PRMs, SOLAR-RL proposes a trajectory-aware reward shaping mechanism. We mathematically fuse global trajectory constraints with retroactive outcome analysis, constructing dense, variation-based reward signals that offer a lightweight yet effective solution to the sparsity problem in semi-online settings.

3 Methodology

In this section, we introduce **SOLAR-RL**, a Semi-online Reinforcement Learning framework designed to address the credit assignment problem in long-horizon GUI navigation. We first formulate the GUI navigation task as a POMDP. Then, we detail our two core components: (1) **Offline Trajectory Reconstruction**, which simulates online interaction dynamics using static data to expand the exploration space; and (2) **Trajectory-Aware Reward Shaping**, which retroactively propagates outcome variations to accurately attribute credit in sparse-reward environments.

3.1 Problem Formulation

Following prior works such as UI-S1 (Lu et al., 2025b) and VAGEN (Wang et al., 2025), we model the GUI agent's interaction as a POMDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the transition function, $\mathcal{R}$ the reward function, and $\gamma \in (0, 1)$ the discount factor. At each time step t, the agent observes a state s_t (typically a screenshot and view hierarchy) and generates an action $a_t \sim \pi_\theta(\cdot|s_t, h_t)$ based on the interaction history h_t. The environment transitions to s_{t+1} and provides a reward r_t.

Unlike standard Online RL, which suffers from high interaction costs and instability, or Offline RL, which struggles with distribution shift, our approach leverages a static dataset $\mathcal{D}_{offline}$ to approximate the optimal policy. We achieve this by simulating the dynamics of online feedback, allowing the agent to simulate pseudo-online feedback in static data by detecting failure points via per-step validity scoring and applying target-aligned reward shaping for retroactive credit assignment.

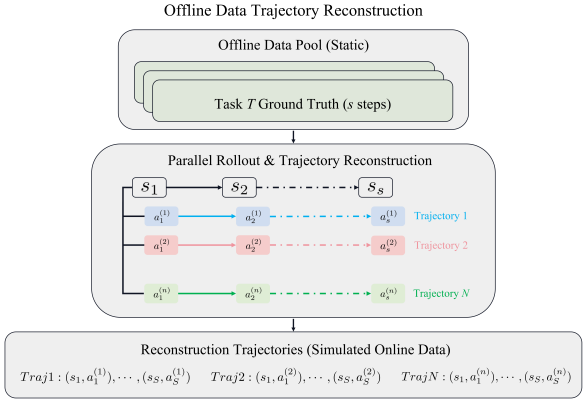


Figure 3: Offline Trajectory Reconstruction. At each step, we run N parallel rollouts and connect candidates with the same rollout index to form N reconstructed trajectories, yielding simulated online data for training.

3.2 Offline Trajectory Reconstruction

To bridge the gap between offline training and online deployment, we propose a trajectory reconstruction mechanism, illustrated in Figure 3, that selectively extends valid rollout branches. For a given task, we perform N independent rollouts at each step t, indexed by $i \in \{1, \ldots, N\}$. Responses with the same index i are chained to form a potential trajectory candidate τ_i. Crucially, although full rollouts are generated, a trajectory τ_i is retrospectively truncated at the first invalid step t^* if the action $a_{t^*}^{(i)}$ is found to be invalid, discarding subsequent steps.

Validity Assessment. To ensure the reconstructed trajectories maintain semantic consistency with expert demonstrations, we employ a rigorous verification protocol using ground-truth (GT) labels. We categorize the supported GUI actions into four groups and apply specific metrics (e.g., Gaussian kernels for coordinates, F1-scores for text). The full criteria are provided in Appendix B.2 (Table 4). This validity filter prunes low-quality deviations while expanding static data into diverse semi-online trajectories.

3.3 Trajectory-Aware Reward Shaping

A core contribution of SOLAR-RL is the Trajectory-Aware Reward Shaping mechanism (as illustrated in Figure 2), which retroactively propagates outcome variations to accurately attribute credit in sparse-reward environments.

Atomic Action Scoring. Before applying reward shaping, we quantify the execution quality of each step using a fine-grained scoring function $\Phi(a_{pred}, a_{GT}) \in [0, 1]$. We map different action primitives (e.g., Click, Scroll, Type) to continuous reward signals.

The complete scoring functions and formulas for all atomic actions are detailed in Appendix B.2 (Table 5).

3.3.1 Trajectory-Level Reward

The trajectory-level reward R_{traj} assesses the overall quality of a reconstructed trajectory. It is designed to encourage not just task completion, but also trajectory completeness and high execution quality. The formulation is defined as:

$$R_{traj} = \frac{\sum_{t=0}^{T} s_t^{\text{raw}}}{T} + \frac{T}{N_{ref}} + \mathbb{I}(Success) \quad (1)$$

where s_t^{raw} is the raw validity score of step t, T is the current trajectory length, N_{ref} is the reference expert length, and $\mathbb{I}(Success)$ is the binary task completion indicator. This reward R_{traj} is assigned to every step t within the trajectory. By incorporating the term $\frac{T}{N_{ref}}$, we explicitly model the trade-off between trajectory progress and task complexity, ensuring that local actions remain aligned with the global objective.

3.3.2 Step-Level Reward with Target Alignment

Standard step rewards often fail in long-horizon tasks due to credit assignment ambiguity and inconsistent magnitudes. To address this, we implement a Target-Aligned Reward Shaping mechanism.

Breakdown Step and Prefix Credit. Due to the trajectory truncation in Section 3.2, each reconstructed trajectory has a first breakdown step t^* where the predicted action becomes invalid under our validity criteria. We treat steps $\{0, \ldots, t^* - 1\}$ as the *valid prefix* and only assign positive credit within this prefix. Invalid steps (including the breakdown step) are penalized to provide explicit corrective signals.

Base Score Calculation. First, we convert the raw validity score $s_t^{\text{raw}} \in [0, 1]$ into a signed base score s_t. Valid actions retain their positive score, while invalid actions are penalized as $-(1 - s_{raw})$. This ensures that the agent receives explicit negative feedback for deviations.

Target Alignment Process. The shaping process then harmonizes these local scores with a global budget in three phases:

1. **Aggregation:** We compute the aggregate positive score over the valid prefix and the aggregate absolute negative score:

$$S_{\text{pos}} = \sum_{t<t^*,\, s_t>0} s_t, \qquad S_{\text{neg}} = \sum_{s_t<0} |s_t|.$$

2. **Base Normalization:** We calculate a normalized reward r_t^{base}. For negative steps (errors), we impose a length-aware dynamic penalty to suppress "reward farming" in long sequences:

$$r_t^{\text{base}} = -\left(\frac{|s_t|}{S_{\text{neg}} + \epsilon} + \lambda \frac{n_{\text{err}}}{\overline{T}}\right), \quad \text{if } s_t < 0 \quad (2)$$

where λ is the penalty coefficient, n_{err} is the count of negative steps, and $\overline{T}$ is the batch average length. For positive steps in the valid prefix, we perform initial normalization: $r_t^{base} = s_t/(S_{pos} + \epsilon)$ for $t < t^*$ and $s_t > 0$.

3. **Total Reward Alignment:** Finally, we enforce a global constraint. We dynamically define the target total reward R_{target} based on the trajectory's global quality (Eq. 1). We then compute the reward gap $\Delta = R_{target} - \sum_t r_t^{base}$ and redistribute it equally among positive steps in the valid prefix ($n_{pos} = \sum_t \mathbb{I}[t < t^* \wedge s_t > 0]$):

$$r_t^{final} = \begin{cases} r_t^{base} + \frac{\Delta}{n_{pos}} & \text{if } t < t^* \wedge s_t > 0 \\ r_t^{base} & \text{otherwise} \end{cases} \quad (3)$$

Model	Android Control (Low)		Android Control (High)		GUI-Odyssey		Android World	Online Data	Training Data	
	TM	SR	TM	SR	TM	EM	SR	Online?	#Steps	#Trajs
I. Generalist Foundation Models *(not specifically optimized for GUI control)*										
Qwen2.5-VL-7B	**94.61**	**85.05**	**73.46**	**61.40**	61.89	47.92	–	–	–	–
GLM-4.1V-Thinking	86.09	80.66	67.31	53.02	72.76	42.57	41.7	–	–	–
GLM-4.5V (106B)	86.35	81.37	71.54	59.15	**75.33**	**48.90**	**57.0**	–	–	–
II. Online Specialized Agents *(Uses environment interaction for data collection and iterative training)*										
GUI-Owl-7B	91.05	86.25	81.60	72.66	81.58	65.22	**66.4**	Y	–	–
UI-TARS-7B-SFT	98.08	94.81	85.00	77.99	86.94	68.82	33.3	Y	–	145K
UI-TARS-72B-SFT	**98.17**	**95.05**	**86.17**	**79.37**	**89.80**	**72.27**	46.6	Y	–	145K
III. Offline Specialized Agents *(Static data only, No interaction costs)*										
AgentCPM-GUI-8B	92.80	**88.60**	76.40	67.93	**90.82**	**74.84**	–	N	>470K	>55K
UI-Venus-Navi-7B	92.17	86.16	79.05	68.61	87.30	71.09	**49.1**	N	350K	–
Aguvis-72B	–	84.4	–	66.4	–	–	26.1	N	–	35K
SOLAR-RL (Ours)	**93.24**	88.57	**79.19**	**69.27**	87.60	68.20	33.7	**N**	**94K**	**15K**

Table 1: Unified comparison on Android Control, GUI-Odyssey, and AndroidWorld. TM/SR denote Type Match and Step Success Rate on Android Control (Low/High); GUI-Odyssey reports Type Match (TM) and Exact Match (EM); AndroidWorld reports Success Rate (SR). Models are grouped by training paradigm: (I) generalist foundation models, (II) agents trained with online-collected interaction trajectories, and (III) offline agents trained on static data only. **Online?** indicates whether environment interaction is used for trajectory collection during training; **#Steps/#Trajs** report training data scale when available ("–" means not reported or not evaluated).

This mechanism aligns the total return with trajectory-level quality while concentrating positive credit on *pre-breakdown* decisions, thereby stabilizing long-horizon optimization under sparse feedback.

4 Experiments

In this section, we rigorously evaluate SOLAR-RL across three diverse benchmarks. We aim to answer two key questions: (1) Does SOLAR-RL achieve competitive performance against state-of-the-art GUI agents, particularly under strict offline constraints? (2) Does the proposed semi-online mechanism effectively mitigate the training instability and policy collapse often observed in standard RL baselines?

4.1 Experimental Setup

Benchmarks. We select three representative benchmarks that cover the full spectrum of GUI agent capabilities. For SOLAR-RL, we report the mean over 4 independent runs (different random seeds). Our evaluation follows the benchmarking framework proposed in guievalkit, and we directly adopt the reported evaluation results of other models provided by this framework:

- **Android Control** (Li et al., 2024): A widely used static benchmark for evaluating atomic action execution. It is divided into *Low* (simple instruction following) and *High* (requiring multi-step reasoning) splits. We report Type Match (TM) and Step Success Rate (SR).
- **GUI-Odyssey** (Lu et al., 2025a): A dataset designed for long-horizon cross-app navigation, featuring complex workflows that span multiple applications. This benchmark tests the agent's ability to maintain context over extended trajectories.
- **Android World** (Rawles et al., 2024): A dynamic benchmarking environment designed to evaluate autonomous agents in real-world Android applications. Unlike static benchmarks, it employs parameterized task generation and durable, system-state-based reward signals. This benchmark serves as the most rigorous test for deployment-time reliability.

Baselines. We compare SOLAR-RL with leading baselines categorized into three groups: **(I) Generalist MLLMs** like Qwen2.5-VL (Bai et al., 2025) and GLM-4.5V (Hong et al., 2025b); **(II) Online Specialized Agents** like UI-TARS (Qin et al., 2025) and GUI-Owl (Ye et al., 2025), which benefit from environment interaction; and **(III) Offline Specialized Agents** like AgentCPM (Zhang et al., 2025), UI-Venus (Gu et al., 2025), and Aguvis (Xu et al., 2024), which share our constraint of learning from static data.

Implementation Details We implement SOLAR-RL on top of the verl framework, initialize the

policy from Qwen2.5-VL-7B-Instruct and train on 15k static trajectories (Appendix A). Trajectory reconstruction uses temperature 1.0 with $N = 8$ candidate rollouts per step. Training runs on 32 NVIDIA L40S GPUs with a global batch size of 128 and a maximum context length of 6,144 tokens, taking **60 hours** for **650 training steps**.Detailed hyperparameters and rollout-engine settings are provided in Appendix B.1.

4.2 Main Results

4.2.1 Fine-Grained Grounding and Control

We first evaluate fundamental grounding capabilities on Android Control. As shown in Table 1 (Category III), SOLAR-RL demonstrates dominant performance among offline agents.

While large-scale models benefit from massive parameter counts, SOLAR-RL achieves **93.24%** Type Match and **88.57%** Success Rate on the *Low* split, ranking second only to AgentCPM-GUI-8B (88.60%) by a negligible margin. Crucially, on the *High* split which requires multi-step reasoning, our method achieves the **highest performance** in the offline category (69.27% SR), outperforming both UI-Venus (68.61%) and AgentCPM (67.93%). This indicates that our trajectory-aware credit assignment effectively prevents reasoning degradation in complex tasks, a common issue in smaller offline models.

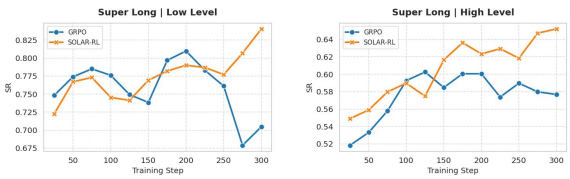
4.2.2 Long-Horizon Navigation

Table 1 highlights the impact of our approach on extended interactions in GUI-Odyssey. Standard offline methods often suffer from compounding errors. By simulating online feedback, SOLAR-RL maintains robust performance, achieving **87.60%** Type Match, effectively bridging the gap between step-level supervision and trajectory-level goals. While AgentCPM shows slightly higher raw metrics here, it relies on a training set that is over **3x larger** (>55k trajectories vs. our 15k), further underscoring the sample efficiency of our approach.

4.2.3 Real-World Execution

We further evaluate SOLAR-RL on Android World, the most dynamic and challenging benchmark. The results in Table 1 reveal a significant advantage in data efficiency.

Within the Offline Category (III), SOLAR-RL achieves a success rate of **33.7%**, securing the second-best position. While UI-Venus achieves a higher score (49.1%), it is trained on 350k steps of data—nearly **4 times** the volume used by SOLAR-RL (94k steps). Moreover, when compared to Online Category (II) baselines, SOLAR-RL outperforms UI-TARS-7B-SFT (33.3%) without requiring any expensive online interaction or the massive 145k trajectory dataset used by the latter. This result exposes a crucial insight: raw data scale is not the only path to performance. By refining the learning signal via trajectory-aware reward shaping, SOLAR-RL achieves competitive real-world robustness with a fraction of the data budget (≈10% of baselines), offering a far more scalable solution for autonomous GUI navigation.



(a) Low Level | Super Long (b) High Level | Super Long

Figure 4: Direct-training ablation on Super Long trajectories. SOLAR-RL avoids late-stage collapse in GRPO and yields higher accuracy, with a larger gain on High-Level tasks.

4.3 Ablation: Efficacy in Direct Training

To deconstruct the contribution of our reward shaping mechanism independent of the two-stage curriculum, we conduct an ablation study involving direct RL training. We compare two methods: standard **GRPO** (using sparse trajectory rewards) and **SOLAR-RL** (using our proposed trajectory-aware reward shaping), both trained from scratch without the first-stage atomic adaptation. We partition the *Android Control* validation set into "Long" ($L \in [6, 13]$) and "Super Long" ($L \geq 14$) buckets based on dataset statistics (see Appendix B.5). Figure 4 visualizes the training dynamics on the most challenging "Super Long" tasks.

Resilience in Long Horizons. The divergence shown in Figure 4 highlights the structural fragility of sparse-reward optimization. Specifically, in the Low-level split (Figure 4a), the baseline GRPO fails to sustain optimization after 200 steps, indicating that sparse terminal signals are insufficient to correct errors in early parts of long trajectories, leading to degenerate loops. In contrast, SOLAR-RL leverages retroactive credit assignment to provide dense feedback, ensuring monotonic policy improvement. This stability advantage is further amplified in High-level tasks (Figure 4b), where

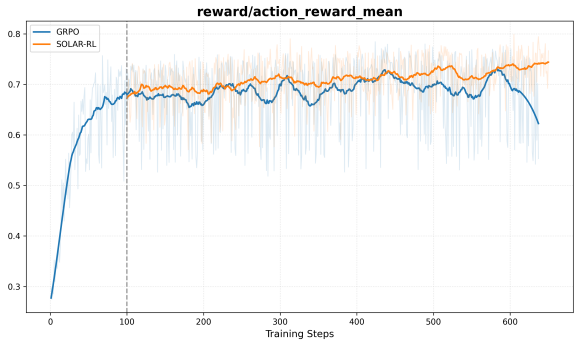


Figure 5: Mean action reward during training. GRPO collapses in later stages, whereas SOLAR-RL improves monotonically and converges.

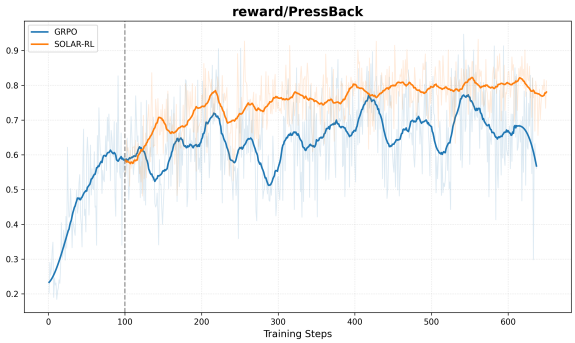
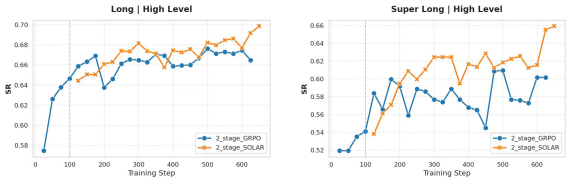


Figure 6: Training dynamics of *PressBack*. SOLAR-RL converges faster and more stably than GRPO, effectively preventing navigation loops.



(a) High Level | Long (b) High Level | Super Long

Figure 7: Two-Stage Training Dynamics on High-Level Tasks. SOLAR-RL improves more steadily and reaches higher accuracy than GRPO.

SOLAR-RL effectively navigates complex reasoning paths that stall standard RL methods. (Similar trends in the "Long" bucket are detailed in Appendix Figure 11.)

4.4 Analysis: Training Dynamics and Stability

Having verified the efficacy of our reward mechanism in direct training, we now analyze the training dynamics of the full **Two-Stage SOLAR-RL** pipeline (Atomic Adaptation $\rightarrow$ Trajectory Optimization). While the main results demonstrate competitive final performance, the core contribution of SOLAR-RL lies in its **training stability** and **sample efficiency**. Standard RL methods like GRPO often suffer from high variance and *policy collapse* in long-horizon tasks. In this section, we provide a deep dive into the learning dynamics to validate our design choices.

4.4.1 Training Stability

We compare SOLAR-RL against the strong baseline, GRPO, to understand why our two-stage pipeline is more stable. Figure 5 shows the evolution of the mean action reward. GRPO improves early but suffers catastrophic degradation after $\sim$600 steps, a typical *policy collapse* failure mode in long-horizon RL where sparse rewards provide inconsistent gradients and encourage overfitting to local optima (e.g., loops). In contrast, SOLAR-RL exhibits monotonic improvement and plateaus at a higher reward (≈ 0.75), suggesting that the two-stage curriculum with trajectory-aware credit assignment and reward shaping stabilizes long-term optimization.

To localize the source of this stability, we further inspect action-level learning. Figure 6 highlights *PressBack*, a critical primitive for error correction.The GRPO baseline exhibits severe oscillation, indicating difficulty in learning when to retreat. SOLAR-RL, leveraging retroactive feedback, quickly converges to high precision (> 0.8), confirming that our method injects global trajectory context into atomic action learning and mitigates the "forgetting" of skills during trajectory-level optimization.

Generalization to Other Primitives. Similar stabilization trends hold across other primitives; detailed per-primitive curves and discussions are provided in **Appendix B.3** and **Figure 8**.We further provide a qualitative failure-case study on continuous decision recovery in Appendix B.4.

4.4.2 Long-Horizon Resilience

Finally, we investigate whether the benefits of SOLAR-RL persist in the most challenging long-horizon scenarios even after Atomic Adaptation. We focus on the **High-Level** split across the *Long* and *Super Long* horizons.

As shown in Fig. 7, SOLAR-RL remains robust under the most challenging long-horizon settings. On the **Long** horizon (Fig. 7a), the 2-stage GRPO baseline quickly saturates around 0.66–0.67, whereas 2-stage SOLAR-RL continues to improve and reaches ≈ 0.70. The advantage be-

comes more pronounced on the **Super Long** horizon (Fig. 7b): the 2-stage GRPO baseline oscillates and largely plateaus around $\approx$ 0.58–0.60, suggesting that a good initialization alone is insufficient for stable long-horizon optimization. In contrast, 2-stage SOLAR-RL continues to improve and reaches a peak SR of $\approx$ 0.66, confirming that trajectory-aware reward shaping provides critical dense guidance for long-term planning beyond atomic adaptation.

5 Conclusion

In this paper, we presented **SOLAR-RL**, designed to bridge the gap between static offline datasets and dynamic online decision-making for GUI agents. Addressing the limitations of myopic offline cloning and sample-inefficient online interaction, we introduced two core mechanisms: Offline Trajectory Reconstruction to simulate diverse interaction pathways, and Trajectory-Aware Reward Shaping to solve the credit assignment problem in sparse-reward environments. Extensive evaluations on Android Control, GUI-Odyssey, and Android World demonstrate that SOLAR-RL improves robustness in *long-horizon* tasks, preventing the policy collapse often observed in baselines.

6 Limitations

While SOLAR-RL effectively bridges the gap between offline stability and online exploration for GUI agents, several limitations remain.

First, the **Semi-Online mechanism is bounded by the coverage of the offline dataset**. Although our trajectory reconstruction expands the exploration space by synthesizing diverse interaction paths, the agent still cannot encounter states or system dynamics that are entirely absent from the source data distribution, such as unseen pop-ups, latency-induced interface shifts, or rare app-specific workflows. In this sense, SOLAR-RL simulates online feedback, but it does not replace real interaction with a live environment.

Second, the **current instantiation relies on ground-truth signals for trajectory validity checking**. In our experiments, the validity filter in Section 3 uses expert annotations and heuristic comparisons (e.g., coordinate distance, text matching, and action-type consistency) to identify the first breakdown step in a clean and controlled manner. However, SOLAR-RL is agnostic to how the validity score is obtained: it can come from (i) ground-truth labels, as in this work, (ii) a learned verifier trained on labeled offline data to predict action validity from observation-action pairs, or (iii) a reward or critic model, such as process reward models or UI verifiers, when explicit labels are unavailable. This suggests a feasible path toward scaling SOLAR-RL to weakly labeled or unlabeled GUI data. At the same time, replacing ground-truth supervision with learned validity estimators introduces new challenges, including reward noise, calibration drift, and reward hacking, which require further investigation.

Third, our experimental evaluation is primarily **concentrated on mobile (Android) environments**. Although trajectory-aware reward shaping is platform-agnostic in principle, extending it to desktop operating systems and web browsers is non-trivial. These environments introduce richer action primitives (e.g., hover, right-click, keyboard shortcuts, drag-and-drop, multi-window interactions, and tab switching), more asynchronous interface changes, and more complex view hierarchies than mobile benchmarks. A faithful extension therefore requires not only broader benchmarks, but also platform-specific validity criteria, curated offline trajectories, and standardized training/evaluation protocols. We leave this cross-platform extension to future work.

References

Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Zhangxuan Gu, Zhengwen Zeng, Zhenyu Xu, Xingran Zhou, Shuheng Shen, Yunfei Liu, Beitong Zhou, Changhua Meng, Tianyu Xia, Weizhi Chen, and 1 others. 2025. Ui-venus technical report: Building high-performance ui agents with rft. *arXiv preprint arXiv:2508.10833*.

Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, and 1 others. 2025a. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang,

Yuxiao Dong, Ming Ding, and 1 others. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, and 1 others. 2025b. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024. On the effects of data scale on computer control agents. *arXiv e-prints*, pages arXiv–2406.

Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. 2025. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*.

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

Zhengxi Lu, Jiabo Ye, Fei Tang, Yongliang Shen, Haiyang Xu, Ziwei Zheng, Weiming Lu, Ming Yan, Fei Huang, Jun Xiao, and 1 others. 2025b. Ui-s1: Advancing gui automation via semi-online reinforcement learning. *arXiv preprint arXiv:2509.11543*.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Christopher Rawles, Sarah Clinckemaillie, 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.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yucheng Shi, Wenhao Yu, Zaitang Li, Yonglin Wang, Hongming Zhang, Ninghao Liu, Haitao Mi, and Dong Yu. 2025. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment. *arXiv preprint arXiv:2507.05720*.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and 1 others. 2026. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, and 1 others. 2025. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. *arXiv preprint arXiv:2510.16907*.

Han Xiao, Guozhi Wang, Yuxiang Chai, Zimu Lu, Weifeng Lin, Hao He, Lue Fan, Liuyang Bian, Rui Hu, Liang Liu, and 1 others. 2025. Ui-genie: A self-improving approach for iteratively boosting mllm-based mobile gui agents. *arXiv preprint arXiv:2505.21496*.

Han Xiao, Guozhi Wang, Hao Wang, Shilong Liu, Yuxiang Chai, Yue Pan, Yufeng Zhou, Xiaoxin Chen, Yafei Wen, and Hongsheng Li. 2026. Ui-mem: Self-evolving experience memory for online reinforcement learning in mobile gui agents. *arXiv preprint arXiv:2602.05832*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024. Aguvis: 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*.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, and 1 others. 2025. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*.

A Data Statistics

Table 2 summarizes the statistics of the open-source data sources and the specific subset selected for training our policy.

Dataset	Android Control	GUI-Odyssey	GUI-Act	Selected for Training
# Steps	84k	114k	15k	94k
# Trajectories	13k	7k	4k	15k

Table 2: Dataset statistics. "Selected for Training" denotes the 15k filtered high-quality trajectories used for initial policy optimization.

B Implementation Details and Supplementary Materials

B.1 Detailed Training Configuration

Table 3 reports the detailed training configuration used in our experiments. Unless otherwise noted, GRPO and SOLAR-RL share the same initialization, rollout-engine settings, optimization hyperparameters, and training budget; the primary difference lies in the reward definition.

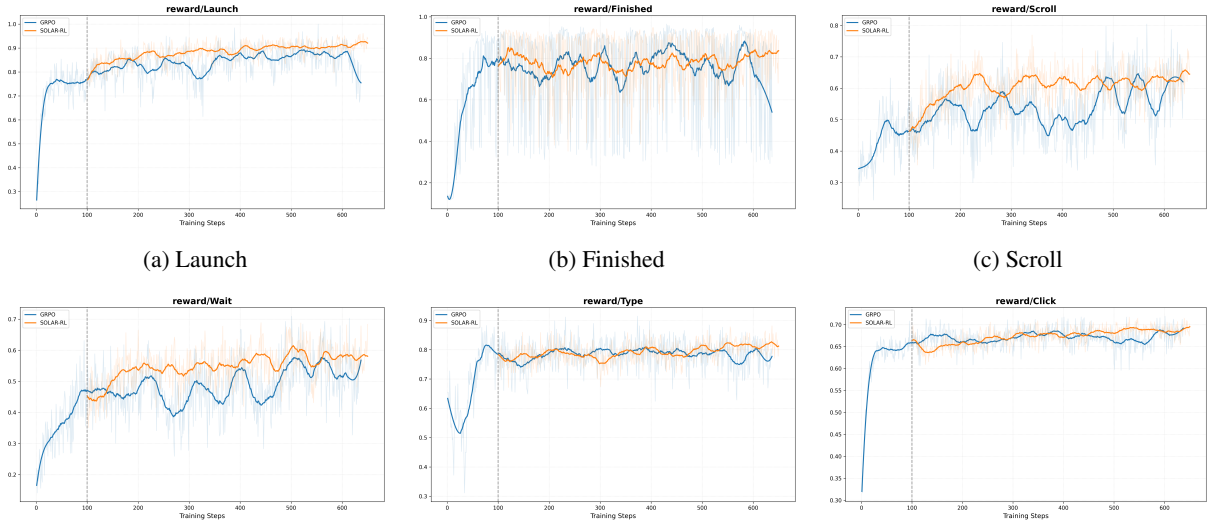
B.2 Methodological Definitions

This section details the criteria and scoring functions (Tables 4, 5) and analyzes the training dynamics shown in Figure 8.

B.3 Analysis of Remaining Action Primitives

Figure 8 summarizes training dynamics for the remaining six action primitives. Consistent with *PressBack*, SOLAR-RL exhibits improved stability across diverse action types. Notably, the gains are most pronounced for primitives that can trigger irreversible state changes (e.g., *Launch*) or amplify downstream compounding errors (e.g., *Scroll* and *Finished*). This trend aligns with SOLAR-RL's design: once a rollout enters an invalid state, failure-point based credit assignment prevents misleading rewards from propagating to earlier steps, and target-aligned shaping further stabilizes the return scale across trajectories. In contrast, primitives with denser local supervision (e.g., *Click*) naturally leave less room for improvement, leading to smaller margins. Overall, these results suggest that SOLAR-RL primarily improves long-horizon credit assignment rather than merely smoothing short-horizon rewards.

- **System transitions (Launch).** In Figure 8a, SOLAR-RL achieves comparable or higher reward with lower variance, indicating more reliable state-switch decisions.
- **Termination (Finished).** In Figure 8b, the baseline collapses late in training (reward drop), while SOLAR-RL remains robust, indicating better long-horizon termination awareness.
- **Long-horizon control (Scroll).** Figure 8c shows reduced oscillation for SOLAR-RL, suggesting fewer redundant/overshooting scrolls that can derail downstream steps.
- **Temporal primitive (Wait).** Figure 8d shows smoother learning for SOLAR-RL, consistent with discouraging unproductive waiting and idle loops.



(a) Launch (b) Finished (c) Scroll

(d) Wait (e) Type (f) Click

Figure 8: Supplementary training curves for the six remaining action primitives. SOLAR-RL (Orange) demonstrates consistent stability advantages over GRPO (Blue).

Category	Hyperparameter	Value
Initialization	Base model	Qwen2.5-VL-7B-Instruct
Rollout	Temperature / N	1.0 / 8 candidates per step
Context	Max context length / max response length	6144 / 1024
Optimizer	Actor learning rate / γ	1×10^{-6} / 0.95
Optimizer	PPO clip (`clip_ratio_low/high`)	0.2
Regularization	KL coefficient / KL loss type	0.001 / `low_var_kl`
Budget	Train batch size / update steps	Global batch size = 128; 650 update steps ($\sim$60h)
Compute	GPUs	32$\times$ NVIDIA L40S
Rollout/Engine	`ppo_max_token_len_per_gpu`	16384
Rollout/Engine	`log_prob_max_token_len_per_gpu`	25600
Rollout/Engine	`max_num_batched_tokens`	32768
Rollout/Engine	`limit_images`	1
Rollout/Engine	`rollout.name`	`vllm`
Rollout/Engine	`VLLM_ATTENTION_BACKEND`	`XFORMERS`

Table 3: Detailed training configuration used for SOLAR-RL and the matched GRPO baseline.

Category	Action	Validity Criteria ($\mathcal{C}_{valid}$)
Coord.	Click/Long	$\text{Dist}(p_{\text{pred}}, p_{\text{gt}}) < \epsilon_{\text{pos}}$
Hybrid	Scroll	$\text{Dist}(p_s) < \epsilon_{pos} \wedge \text{Dir}_{\surd}$
Text	Type	$\text{F1}(\text{txt}_{pred}, \text{txt}_{gt}) > \delta_{text}$
	Launch	$\text{Sim}(\text{app}_{pred}, \text{app}_{gt}) > 0.9$
System & Ctrl	Wait/Back Home/etc.	$\mathbb{I}(\text{Type}_{pred} = \text{Type}_{gt})$

Table 4: Validity assessment criteria.

Action	Metric	Scoring Function (s_{raw})
Click	Spatial	$e^{-\frac{\|p_{pred}-p_{gt}\|^2}{2\sigma^2}}$
Scroll	Hybrid	$e^{-\frac{\|p_s-\hat{p}_s\|^2}{2\sigma^2}} \cdot \mathbb{I}(\text{Dir}_{\surd})$
Type	F1	$2 \cdot \frac{\text{Pre}\cdot\text{Rec}}{\text{Pre}+\text{Rec}}$
Launch	Threshold	$\mathbb{I}(\text{Sim}(\text{App}_{pred}, \text{App}_{gt}) > 0.9)$
System	Exact	$\mathbb{I}(\text{Type}_{pred} = \text{Type}_{gt})$

Table 5: Atomic action scoring mechanism.

- **Text-conditioned (Type).** In Figure 8e, SOLAR-RL is steadier, suggesting reduced reward noise for partially-correct typings.
- **Fine-grained pointing (Click).** Figure 8f shows similar or slightly better reward with lower variance; gains are smaller since click is more densely supervised.

B.4 Failure Case Analysis: Continuous Decision Recovery in Long-Horizon GUI Tasks

Figure 9 presents a representative case from training on *Simple SMS Messenger*. The task is to resend the most recent message to a specific contact. This example highlights that long-horizon success does not merely depend on avoiding the first mistake; rather, it depends on whether the agent can recover repeatedly after entering suboptimal states.

In Trajectory 1, the agent first makes an incorrect context-level decision by attempting a long-press in the conversation list instead of entering the chat thread. It then partially recovers by opening the correct thread, but fails to execute the second recovery: after realizing that long-pressing the sent message is ineffective, it does not switch to the correct alternative strategy of re-typing and re-sending the message. As a result, the rollout remains trapped in a locally plausible but globally unsuccessful behavior pattern.

Trajectory 2 follows a similar early prefix but successfully performs two consecutive recoveries. The agent first switches to the correct interaction context (conversation list $\rightarrow$ chat thread), and then revises the action strategy (ineffective long-press $\rightarrow$ re-type-and-send). This leads to successful task completion. The comparison shows that long-horizon GUI tasks require *continuous decision recovery*: correcting one mistake is often insufficient if the agent cannot adapt again when the next local strategy fails.

This phenomenon is consistent with the design motivation of SOLAR-RL. Failure-point based credit assignment helps localize where the rollout first becomes invalid, while trajectory-aware reward shaping discourages persistent but unproductive action chains after the breakdown point. Qualitatively, this encourages the policy to abandon ineffective local behaviors and re-enter a valid

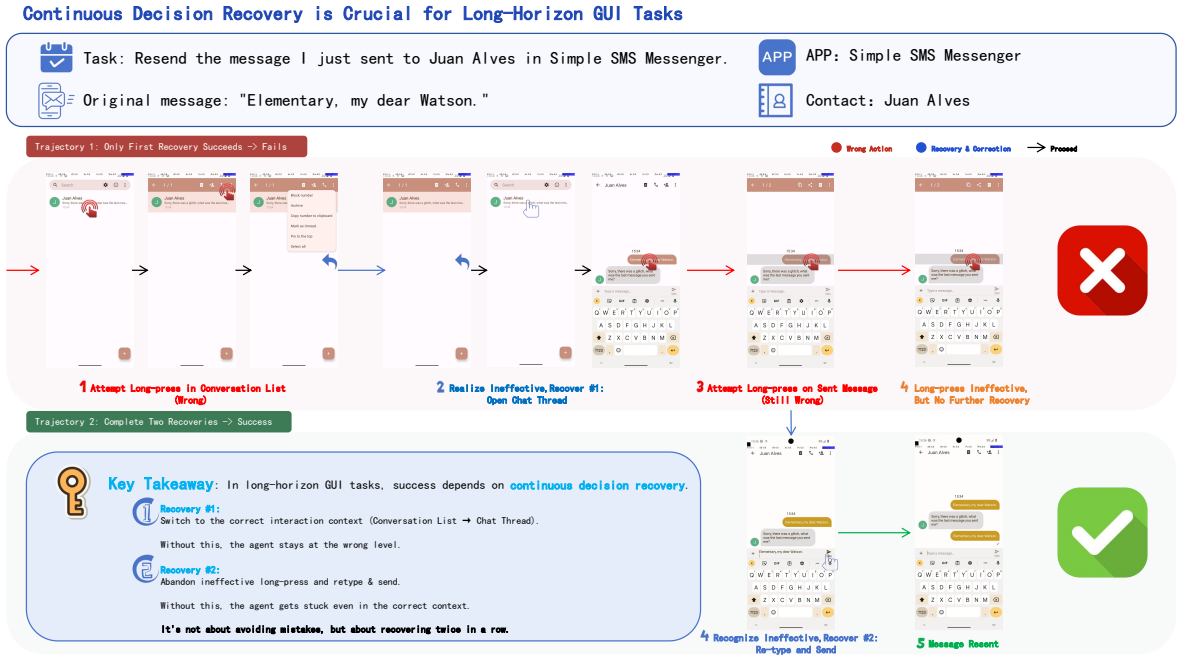


Figure 9: A qualitative failure-case study on continuous decision recovery. Both trajectories start from the same task, but only the second one succeeds because the agent performs two consecutive recoveries: it first switches to the correct interaction context and then replaces an ineffective long-press strategy with re-type-and-send. The example illustrates that in long-horizon GUI tasks, robustness depends not only on avoiding errors, but on recovering from multiple errors in sequence.

decision path instead of repeating superficially similar actions.

B.5 Detailed Ablation on Trajectory Lengths

In Section 4.3, we partitioned tasks based on the statistical distribution of optimal path lengths in the training data. Figure 10 presents the histogram and boxplot of these lengths. Based on the quartiles ($Q1 = 4, Median = 6, Q3 = 8$), we define "Long" tasks as the interquartile extension ($L \in [6, 13]$) and "Super Long" tasks as the upper tail ($L \geq 14$).

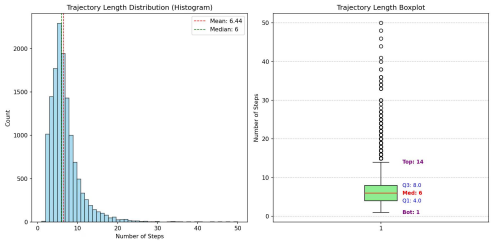
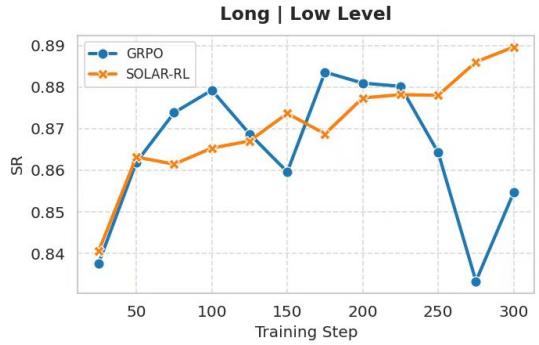
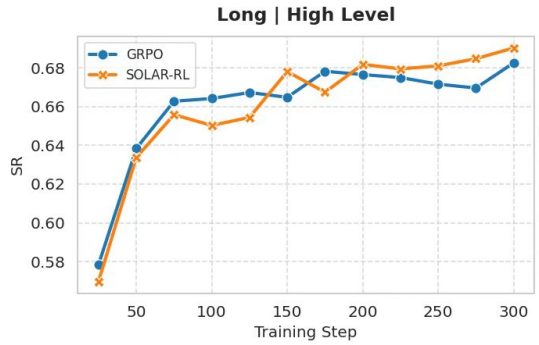


Figure 10: Distribution of trajectory lengths in the training dataset. We use the distribution statistics to define the "Long" and "Super Long" buckets for ablation.

Figure 11 provides the full set of training dynamics comparisons, including the "Long" bucket ($L \in [6, 13]$) which was omitted from the main text. Consistent with the Super-Long results, SOLAR-RL (Orange) demonstrates superior sample efficiency and stability across both difficulty levels (Low/High) in this intermediate length range.

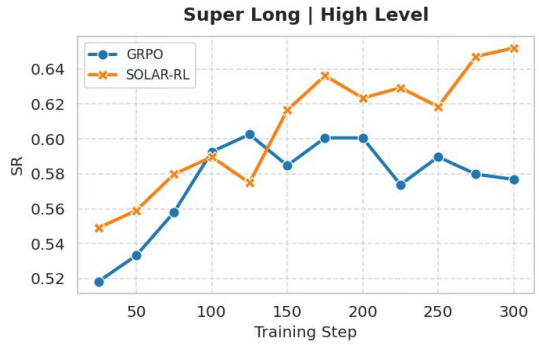


(a) Low Level | Long



(b) High Level | Long

(c) Low Level | Super Long



(d) High Level | Super Long

Figure 11: **Complete direct-training ablation (Action SR).** Long: $L \in [6, 13]$; Super Long: $L \geq 14$. SOLAR-RL consistently outperforms GRPO, with a larger gap on longer and harder horizons.