TRAIN TO VERIFY, DEPLOY TO ACT: EFFICIENT LEARNING FOR MOBILE GUI AGENTS

Xingjian Diao
Dartmouth College
NH, USA

ABSTRACT

Recent advances in vision-language models are turning GUI agents from interface perception and element grounding systems into general agents capable of autonomously executing real-world user tasks. However, long-horizon interaction remains vulnerable to silent action failures, state drift, repetitive behaviors, and inadequate recovery, while frontier agents often mitigate these problems with lengthy reasoning, iterative reflection, or multi-model verification at every step, creating inference latency and computational costs that accumulate with task length and have become a major barrier to practical deployment. We introduce TRAVEC, a framework for transition and recovery alignment via verifiable evidence-grounded contracts, which systematically embeds state abstraction, transition prediction, outcome verification, and failure diagnosis into data construction and reinforcement learning so that the deployed Action Policy only needs to emit executable actions. TRAVEC provides an automated annotation and generation pipeline that unifies heterogeneous GUI trajectories and combines deterministic state differencing, evidence-grounded multimodal annotation, and counterfactual recovery generation to produce transition-contract, negative-recovery, preference, and verifier-calibration supervision at scale. Building on these data, TRAVEC trains two auxiliary adapters, a Contract Policy and a Verifier, uses them to create auditable self-generated experience through resettable predict–execute–verify–recover rollouts, and optimizes the Action Policy with offline GRPO using six rewards: Action Type, Action Argument, and our four proposed rewards for State Delta, Recovery, Progress/Loop, and Online Environment. Starting from Qwen3.5-9B, we train TRAVEC-9B and compare it with seven frontier models on five benchmarks, with extensive ablations further validating its components; the results demonstrate clear advantages in both task performance and inference efficiency.

1 INTRODUCTION

Graphical user interfaces (GUIs) are the primary gateway through which people access mobile applications, desktop software, and web services, and therefore provide a natural substrate for general agents to turn language intentions into consequential digital actions (Wang et al., 2025b; Tang et al., 2025). Foundation-model-based GUI agents combine interface perception, language understanding, task planning, and action generation to execute multi-step instructions using generic operations such as clicking, swiping, and typing (Wang et al., 2025b; Tang et al., 2025). CogAgent and SeeClick showed that directly modeling small interface elements and spatial locations can reduce reliance on HTML, accessibility trees, or application-specific APIs (Hong et al., 2024; Cheng et al., 2024). OS-World and AndroidWorld subsequently moved evaluation from static element localization to reproducible interaction with real operating systems, where cross-application behavior, long trajectories, and execution outcomes can be checked by the environment (Xie et al., 2024; Rawles et al., 2025). Recent foundation GUI agents further share environments, data, and action representations across mobile, desktop, and web platforms, marking a shift from local GUI grounding toward general digital task execution (Tang et al., 2025; Zhou et al., 2026a).

GUI agents have advanced from grounding to decision making and environment-aligned learning. CogAgent and SeeClick established high-resolution grounding, while Mobile-Agent-v2 and Agent S

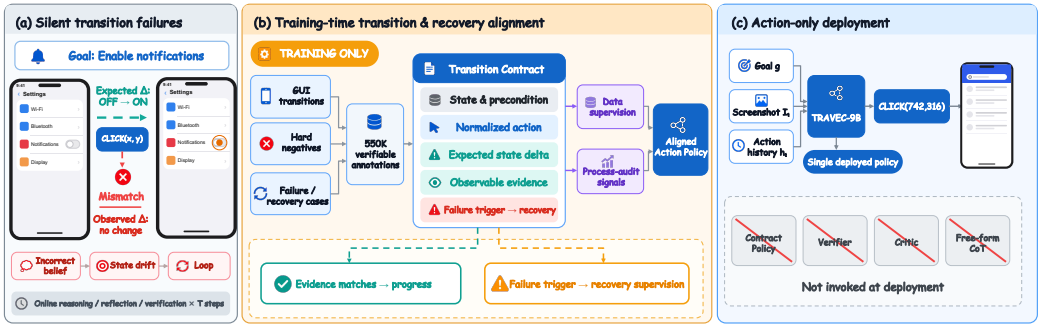

Figure 1: TRAVEC moves transition verification and failure recovery from online reasoning into training. Evidence-grounded contracts and process-audit signals align a single action-only policy, while auxiliary models and free-form reasoning are not invoked at deployment.

added decomposition, memory, retrieval, and reflection for long horizons (Hong et al., 2024; Cheng et al., 2024; Wang et al., 2024; Agashe et al., 2025a). Aguvis and UI-TARS extended cross-platform actions and System-2 reasoning, while ComputerRL and MobileForge brought large-scale online learning and annotation-free real-app adaptation (Xu et al., 2025b; Qin et al., 2025; Lai et al., 2026; Liu et al., 2026b). Two challenges remain: *long-horizon reliability*, because valid-looking actions do not verify state changes and failures accumulate (Rawles et al., 2025; Yu et al., 2026); and *inference efficiency*, because stronger agents add step-wise reasoning and reflection during execution (Qin et al., 2025; Huang et al., 2026).

We move state abstraction, consequence prediction, verification, and failure diagnosis from online reasoning into data supervision and reinforcement-learning signals, as illustrated in Figure 1. *Transition contracts* bind task state, preconditions, normalized actions, expected changes, evidence, and recovery. We unify five GUI datasets and derive page, focus, input, blocking, text-change, and no-change signals. A multimodal annotator generates contracts, hard negatives, counterfactual recoveries, preferences, and verifier-calibration examples filtered by deterministic checks. Post-action observations supervise labels but never enter the Action Policy, yielding 550K leakage-free annotations.

Starting from Qwen3.5-9B, TRAVEC trains an Action Policy that emits exactly one normalized action and independently branches two training-time adapters: a Contract Policy that predicts a transition before execution and a Verifier that evaluates the observed outcome afterward. In resettable Android environments, each task is executed from the same snapshot in four rollouts. The Contract Policy and Verifier provide pre- and post-action checks, while a trajectory Critic assesses task outcome, step reasonableness, and non-oracle recovery hints, forming a predict–execute–verify–recover audit loop. Reliable self-generated steps are used for offline GRPO with Action Type and Action Argument rewards adapted from MobileForge (Liu et al., 2026b), together with four rewards proposed here: State Delta, Recovery, Progress/Loop, and Online Environment. At deployment, the Contract Policy, Verifier, and Critic are removed; TRAVEC-9B directly produces action-only outputs. We evaluate the resulting model on six static and dynamic benchmarks against seven frontier GUI agents and conduct component, data, and reward ablations.

Our contributions are threefold:

- We develop a scalable automated GUI annotation and generation pipeline. Its transition contracts jointly represent task state, preconditions, expected changes, observable evidence, and failure recovery, enabling 550K item-level verifiable annotations through deterministic differencing, evidence-grounded labeling, and counterfactual generation.

- We introduce a training-time process-audit and recovery mechanism built from Contract and Verifier adapters plus a trajectory Critic. It extracts leakage-free offline GRPO data from self-generated interaction and aligns the policy with six rewards covering action protocol, state transition, recovery, task progress, loop avoidance, and real execution outcomes.

- We distill structured reasoning, verification, and recovery into TRAVEC-9B while retaining a single action-only policy at deployment. Evaluation on five benchmarks, seven frontier models, and systematic ablations examines both task performance and inference efficiency.

2 Related Work

2.1 GUI grounding, planning, memory, and recovery

GUI agents must ground language in ambiguous interfaces and execute multi-step plans. Grounding methods improve visual resolution, region awareness, interaction extraction, cross-platform coverage, or visual-token efficiency (Hong et al., 2024; Cheng et al., 2024; You et al., 2024; Lu et al., 2024; Gou et al., 2025; Wu et al., 2025; Lin et al., 2024). Native action models extend cross-platform trajectory modeling (Xu et al., 2025b; Qin et al., 2025; Liu et al., 2026c; Ye et al., 2025; Xu et al., 2026). Long-horizon systems add decomposition, retrieval, reflection, memory, or auxiliary tools (Wang et al., 2024; Li et al., 2025b; Agashe et al., 2025a;b; Qin et al., 2025; Wang et al., 2025a; Zhou et al., 2026b; Liu et al., 2026a; Xiao et al., 2026; Lu et al., 2026). These systems retain auxiliary calls during execution. TRAVEC moves transition and recovery supervision to training and keeps deployment action-only.

2.2 Automated experience, reinforcement learning, and efficient execution

Automated data and environment learning support GUI agents. OS-Genesis, GUI-360°, and Open-Mobile automate synthesis and exploration (Sun et al., 2025; Mu et al., 2025; Cheng et al., 2026). GUI-R1, AgentCPM-GUI, GUI-Libra, and UI-S1 apply rule-based or partially verifiable GRPO offline (Luo et al., 2025; Zhang et al., 2025; Yang et al., 2026; Lu et al., 2025b). ComputerRL, MobileRL, MobileForge, and OS-Themis scale learning or add generated tasks and evidence review (Lai et al., 2026; Xu et al., 2025a; Liu et al., 2026b; Li et al., 2026). Efficiency methods compress history or anticipate decoding (Huang et al., 2025; 2026; Dong et al., 2026). TRAVEC turns transition, evidence, recovery, and verification into offline training signals.

3 Method

TRAVEC places structured transition reasoning and verification in data construction and policy optimization, while deployment retains only the Action Policy. Section 3.1 defines the decision problem and transition contract; Section 3.2 presents the automated data pipeline; Section 3.3 trains three responsibility-separated adapters; and Section 3.4 describes process-audited offline GRPO.

3.1 Problem formulation

We model goal-conditioned GUI interaction as a partially observable decision process $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, \Omega, \mathcal{G}, \mathcal{Y})$. At step t, the environment record $o_t = (I_t, U_t, X_t)$ may contain a screenshot, a UI hierarchy, and OCR or other structured evidence for data construction and auxiliary verification. The deployed Action Policy receives only the screenshot I_t. Given goal g, action history $h_t = (a_1, \ldots, a_{t-1})$, and an optional hint η_t derived only from an earlier failed attempt, it predicts

$$\pi_{\theta_A}(a_t \mid g, I_t, h_t, \eta_t), \qquad a_t = (\alpha_t, \psi_t), \tag{1}$$

where α_t is an action type and ψ_t its type-specific argument. The normalized type set includes click, long press, swipe, type, key/system button, open, wait, and terminate. A trajectory $\tau = (g, o_1, a_1, \ldots, o_T, a_T, o_{T+1})$ receives an execution-based outcome $Y(\tau) \in \{0, 1\}$. We seek $\theta_A^* = \arg\max_{\theta_A} \mathbb{E}[Y(\tau)]$ subject to the deployment constraint that each step emits exactly one parseable action and invokes no auxiliary model.

Let $d_t = \mathcal{D}(o_t, o_{t+1})$ be a deterministic observable state difference containing page/activity, text, focus, input, task-slot, blocking-event, and no-change fields. Rather than predict future pixels, we describe the task-relevant local consequence with a transition contract

$$c_t = (\tilde{s}_t, p_t, a_t, \hat{d}_t, E_t, \rho_t), \qquad \rho_t = (\phi_t, a_t^{\mathrm{rec}}, \hat{d}_t^{\mathrm{rec}}, E_t^{\mathrm{rec}}). \tag{2}$$

Here $\tilde{s}_t$ is a compact task state, p_t the action precondition, $\hat{d}_t$ the expected difference, E_t up to three checkable evidence queries, and ρ_t an optional recovery trigger, action, expected difference, and evidence. During training, a Contract Policy predicts $q_{\theta_C}(c_t \mid g, o_t, h_t, a_t)$ before execution, while a Verifier predicts $V_{\theta_V}(y_t^V \mid g, o_t, c_t, o_{t+1}, d_t, \chi_t)$ afterward. The deterministic checks χ_t cover parsing, coordinates, enumerations, and action consistency. The verifier output $y_t^V = (\mathbf{v}_t, \nu_t, \kappa_t)$ contains eight 0–4 scores, an accept/partial/reject verdict, and 0–4 confidence. Neither auxiliary mapping is deployed.

3.2 Automated Data Annotation and Generation

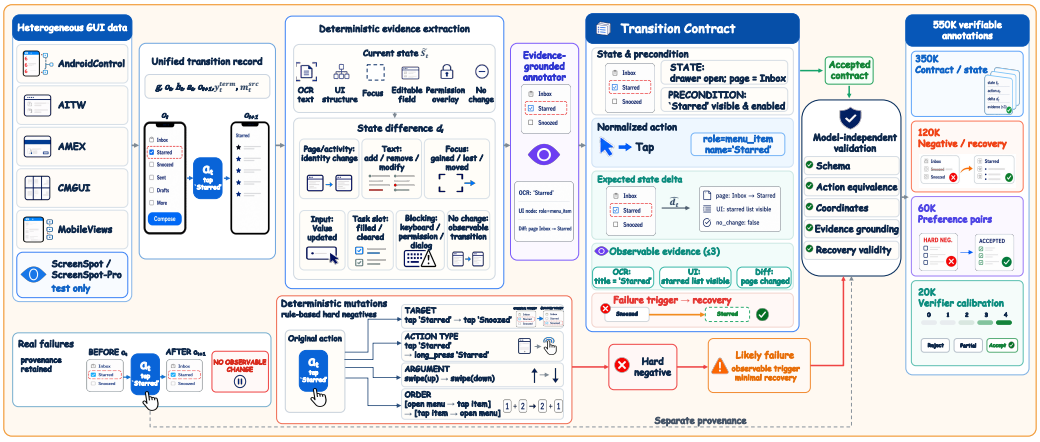

Figure 2: Automated construction of verifiable transition supervision. Deterministic state differencing, evidence-grounded annotation, counterfactual mutation, and model-independent validation yield 550K contract/state, negative/recovery, preference, and verifier-calibration annotations.

Unified transition records. Figure 2 summarizes the construction of verifiable transition supervision. We normalize AndroidControl, AITW, AMEX, CMGUI, MobileViews, ScreenSpot, and ScreenSpot-Pro into records $x_t = (g, o_t, h_t, a_t, o_{t+1}, y_t^{\text{term}}, m_t^{\text{src}})$. Within this annotation pipeline, ScreenSpot and ScreenSpot-Pro are assigned exclusively to the validation and test splits of the Contract Policy and Transition Verifier and are excluded from all training splits. Coordinates are stored in $[0,1]^2$ and deterministically rendered to integer coordinates in $[0,1000]^2$; swipes follow touch-to-lift semantics. These static grounding samples assess localization but are not treated as strong transition evidence. A deterministic extractor first computes a weak current state $\bar{s}_t = F_{\text{state}}(g, o_t)$ and, when a next observation exists, the structured difference $d_t = \mathcal{D}(o_t, o_{t+1})$. Perceptual hashes, OCR, UI structure, page/activity identity, focus, editable fields, keyboard or permission overlays, and task slots jointly support machine-checkable evidence.

The formal pipeline uses Claude Opus 4.8 as the multimodal annotator, but acceptance is controlled by model-independent validators. The target recipe contains 350K contract/state items, 120K negative-recovery items, 60K preference items, and 20K verifier-calibration items, totaling 550K annotations. Before the full run, a stratified 100-item quality gate requires at least 98% process success, 99% schema success, at most 3% evidence error, 2% action-equivalent negatives, 5% invalid recoveries, and at least 95% independent audit pass rate.

Transition-contract annotation. For a demonstrated transition with adjacent observations, an evidence-grounded annotator produces $c_t^+ = A_C(x_t, \bar{s}_t, d_t)$ without rewriting the demonstrated action. An item is retained only if the contract action is semantically equivalent to a_t and the deterministic validator accepts its wrapper, JSON schema, fields, coordinates, evidence cardinality, and recovery structure:

$$\text{Eq}_{\mathcal{A}}(a(c_t^+), a_t) = 1, \qquad V_{\text{det}}(c_t^+, x_t) = 1. \tag{3}$$

Current-observation evidence is used only for $\tilde{s}_t$ and p_t; o_{t+1} and d_t supervise the expected difference and evidence. Thus the next frame never becomes policy input.

Counterfactual negative and recovery generation. For each accepted contract, deterministic mutations construct up to four hard-negative actions by perturbing the target, action type, argument, or order while preserving syntactic plausibility. The annotator predicts the likely failed transition, an observable failure trigger, and a minimal recovery branch. We reject a negative if it is action-equivalent to the demonstrated action, unsupported by the current interface, or paired with a recovery that merely repeats the failed action. Real failed transitions, when available, are retained separately from counterfactual failures so that evidence provenance remains explicit.

Preference and verifier calibration. Preference pairs hold the decision input fixed and contrast an accepted contract or recovery with a hard negative. Calibration items combine positive transitions, counterfactual negatives, real failures, and deterministic corruption cases. They supervise eight verifier dimensions: output format, action validity, state grounding, delta evidence, task progress, recovery applicability, loop safety, and cost efficiency. This design makes the verifier sensitive to realized consequences rather than lexical similarity alone.

3.3 Supervised training of role-specialized adapters

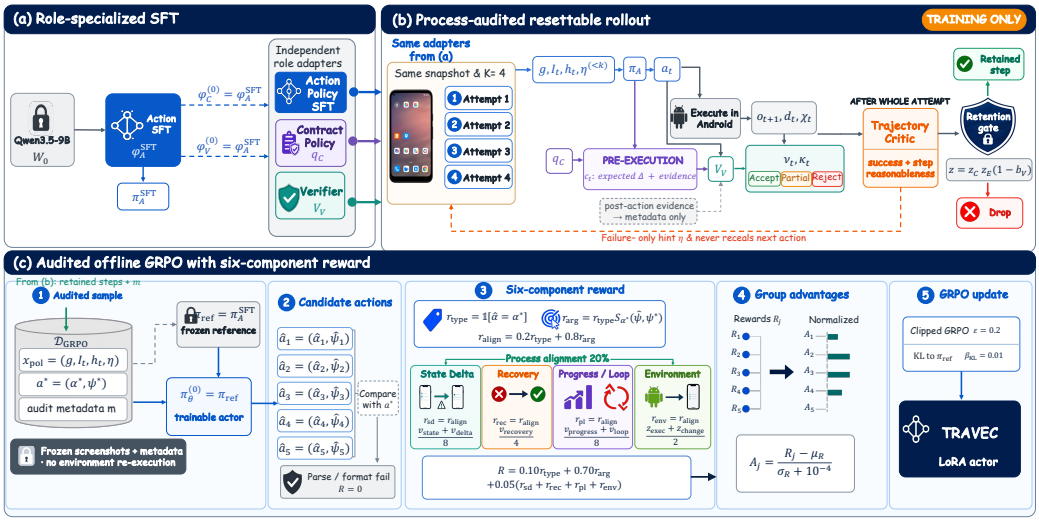

Figure 3: TRAVEC training pipeline. Role-specialized Action, Contract, and Verifier adapters are first trained with supervised learning, then used with a trajectory Critic to audit resettable self-generated rollouts. Retained steps are optimized with offline GRPO using six rewards for action correctness, state transition, recovery, progress/loop behavior, and environment outcomes.

We parameterize each role as $f_{W_0,\phi}$, a Qwen3.5-9B backbone W_0 augmented by one set of LoRA parameters ϕ. Action SFT first learns ϕ_A^{SFT}, defining the action-only policy $\pi_{\theta_A^{\mathrm{SFT}}} = f_{W_0,\phi_A^{\mathrm{SFT}}}$. The Contract and Verifier branches are then initialized from independent copies of ϕ_A^{SFT}:

$$\phi_C^{(0)} = \phi_A^{\mathrm{SFT}}, \qquad \phi_V^{(0)} = \phi_A^{\mathrm{SFT}}. \tag{4}$$

They continue supervised optimization under separate role-specific objectives, yielding $q_{\theta_C} = f_{W_0,\phi_C^*}$ and $V_{\theta_V} = f_{W_0,\phi_V^*}$. Thus the auxiliary branches inherit the interface and action prior learned by Action SFT, but have independent parameters and optimization paths after branching.

Action Policy. The Action Policy generates only `<action>{flat JSON}</action>`. A shared renderer/parser enforces one tag, a fixed field set, type-specific arguments, and coordinate validity. Completion-only cross-entropy is applied to 300K action examples for two epochs at learning rate 2×10^{-5} with rank-16 LoRA. The vision encoder is frozen; LoRA targets language and multimodal projector/merger linear layers.

Contract Policy. Starting from $\phi_C^{(0)}$, the Contract Policy continues SFT on expert contracts, real recoveries, and counterfactual recoveries with loss weights $(0.70, 0.20, 0.10)$. It conditions on

(g, o_t, h_t, a_t) but cannot access o_{t+1} or d_t, so every predicted contract is available before action execution. The train/dev/test sets contain 287,840/6,080/6,080 samples; each distributed optimizer batch fixes the three task counts at 156/34/34. Training uses two epochs, learning rate 10^{-6}, rank-16 LoRA, and a 256-token completion limit.

Transition Verifier. The Verifier is independently initialized at $\phi_V^{(0)}$ and continues SFT without passing through the Contract branch. It observes the pre-action state, contract, post-action state, deterministic difference, and checks. Positive, hard-negative, and calibration losses are weighted $(0.45, 0.45, 0.10)$. The formal set contains 77,100 positives, 77,100 hard negatives, and 18,504 calibration samples (172,704 total), with fixed per-batch counts 100/100/24. We use two epochs, learning rate 10^{-6}, rank-16 LoRA, and a 128-token completion limit. Contract and Verifier remain separate models during rollout, preserving their pre-execution prediction and post-execution verification roles.

3.4 PROCESS-AUDITED OFFLINE GRPO

Resettable rollout and audit. The executable pool contains 3,249 tasks from 20 applications. After within-app token-Jaccard deduplication at threshold 0.9, we stratify 400 training and 40 development tasks. Each task is attempted $K = 4$ times from the same clean AndroidWorld snapshot. At step t, the Action Policy proposes a_t from I_t, the Contract Policy predicts c_t before execution, the environment returns o_{t+1}, and the Verifier evaluates the realized transition. After each attempt, a trajectory Critic labels feasibility, final success, and every step's reasonableness, then provides only an observed-failure/avoid-repeating hint for the next attempt; it is forbidden from revealing the correct next action.

Let $z_{i,t}^{C,(k)}$ denote Critic reasonableness, $z_{i,t}^{E,(k)}$ deterministic execution validity, and let a high-confidence verifier veto be $b_{i,t}^{V,(k)} = \mathbf{1}[\nu_{i,t}^{(k)} = \texttt{reject} \wedge \kappa_{i,t}^{(k)} \geq 3]$. A step is retained iff

$$z_{i,t}^{(k)} = z_{i,t}^{C,(k)} z_{i,t}^{E,(k)} (1 - b_{i,t}^{V,(k)}). \tag{5}$$

The verifier can remove a Critic-positive step but never turn a Critic-negative step positive. A verifier parse failure does not veto the step.

Offline sample construction. For task i, we retain tasks whose four-attempt success rate is in $[0, 0.9]$, excluding consistently solved tasks. Among attempts, we select lexicographically by task success, retained-step ratio, and shorter length. Only retained self-generated actions are materialized:

$$\mathcal{D}_{\text{GRPO}} = \{(x_{i,t}^{\text{pol}}, a_{i,t}^*, m_{i,t}) : z_{i,t}^{(k_i^*)} = 1\}. \tag{6}$$

The policy input $x_{i,t}^{\text{pol}} = (g_i, I_{i,t}, h_{i,t}, \eta_i^{(<k_i^*)})$ contains decision-time information only. Contracts, post-action observations, verifier/Critic labels, and execution outcomes remain in metadata $m_{i,t}$ for auditing and reward computation.

Six-component reward. A candidate completion is parsed as $\hat{a} = (\hat{\alpha}, \hat{\psi})$ and compared with retained action $a^* = (\alpha^*, \psi^*)$. Parse or format failure sets all rewards to zero. Following the adaptive action scoring used by MobileForge (Liu et al., 2026b), we separate

$$r_{\text{type}} = \mathbf{1}[\hat{\alpha} = \alpha^*], \qquad r_{\text{arg}} = r_{\text{type}} S_{\alpha^*}(\hat{\psi}, \psi^*), \qquad r_{\text{align}} = 0.2 r_{\text{type}} + 0.8 r_{\text{arg}}. \tag{7}$$

S_{α^*} uses distance-decayed point similarity for clicks/long presses, exact direction for swipes, token-set F1 for typing, exact or containment matching for app names, and exact type-specific arguments otherwise. We propose four additional rewards from audited metadata:

$$r_{\text{sd}} = r_{\text{align}}(v_{\text{state}} + v_{\text{delta}})/8, \qquad r_{\text{rec}} = r_{\text{align}} v_{\text{recovery}}/4, \tag{8}$$

$$r_{\text{pl}} = r_{\text{align}}(v_{\text{progress}} + v_{\text{loop}})/8, \qquad r_{\text{env}} = r_{\text{align}}(z_{\text{exec}} + z_{\text{change}})/2. \tag{9}$$

State Delta checks grounded expected change; Recovery checks applicability to the observed failure; Progress/Loop rewards progress while penalizing repetition; and Online Environment uses actual execution validity plus task success or observable state change. The formal reward is

$$\begin{aligned} R = {} & 0.10 r_{\text{type}} + 0.70 r_{\text{arg}} + 0.05 r_{\text{sd}} \\ & + 0.05 r_{\text{rec}} + 0.05 r_{\text{pl}} + 0.05 r_{\text{env}}. \end{aligned} \tag{10}$$

Table 1: Offline benchmark performance and AndroidControl inference efficiency. AndroidControl reports success rate (SR). SS and Pro denote target-hit accuracy on ScreenSpot and ScreenSpot-Pro, respectively. Lat. denotes the mean inference time per action (s), and Tok. the mean number of generated tokens per action. Best and second-best results are shown in bold and underlined, respectively.

Model	AndroidControl		GUI-Odyssey			ScreenSpot		Efficiency	
	High ↑	Low ↑	TM ↑	GR ↑	SR ↑	SS ↑	SS-Pro ↑	Lat. (s) ↓	Tok. ↓
Aguvis-7B-720P	57.237	66.914	74.137	70.541	53.373	78.852	20.936	2.853	34.505
GUI-R1-7B	22.464	42.436	70.889	58.397	42.590	<u>85.849</u>	28.716	6.224	32.096
ScaleCUA-7B	36.056	46.608	77.218	69.886	57.315	73.899	23.340	3.591	35.800
GUI-Libra-8B	<u>62.856</u>	<u>81.417</u>	63.181	71.155	43.994	85.535	<u>53.700</u>	5.905	78.294
OpenMobile-8B	52.335	72.502	67.164	59.893	42.332	79.324	38.646	5.763	58.644
MemGUI-8B-SFT	58.061	77.424	68.932	62.140	45.444	76.965	39.216	12.910	318.260
ForgeQwen3-8B	62.665	79.850	**79.864**	71.221	<u>58.369</u>	82.704	46.490	7.070	86.799
Qwen3.5-9B	34.379	52.966	62.217	<u>90.823</u>	56.185	38.758	42.631	<u>0.815</u>	<u>18.039</u>
TRAVEC-9B (ours)	**64.309**	**93.337**	<u>79.588</u>	**91.006**	**72.212**	**92.689**	**67.805**	**0.786**	**17.152**

Thus action matching contributes 80% of the reward mass and the four proposed process rewards contribute the remaining 20%.

GRPO and deployment. GRPO continues optimization from the Action SFT policy rather than restarting from the backbone. We freeze the SFT policy as the reference and initialize the trainable actor to the same policy:

$$\pi_{\text{ref}} = \pi_{\theta_A^{\text{SFT}}}, \qquad \pi_{\theta^{(0)}} = \pi_{\text{ref}}. \tag{11}$$

The actor is optimized with rank-64 LoRA while the reference remains fixed. For each prompt, $G = 5$ completions receive group-normalized advantages $A_j = (R_j - \mu_R)/(\sigma_R + 10^{-4})$. We optimize the clipped GRPO objective with $\epsilon = 0.2$ and KL coefficient $\beta_{\text{KL}} = 0.01$:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}\left[\frac{1}{G}\sum_{j=1}^{G}\min(\xi_j A_j, \text{clip}(\xi_j, 1-\epsilon, 1+\epsilon)A_j)\right] + \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \tag{12}$$

Training runs for four epochs at learning rate 10^{-6} on 56 A100 GPUs; an optimizer update contains 112 unique prompts and 560 completions. It is offline because policy optimization reads frozen screenshots and audit metadata rather than re-executing the environment for newly sampled completions. Let θ_A^{TRAVEC} denote the optimized Action Policy. Deployment uses only

$$\pi_{\text{deploy}} \equiv \pi_{\theta_A^{\text{TRAVEC}}}, \tag{13}$$

which emits one normalized action per step. The Contract Policy, Verifier, and Critic are not invoked, and no contract, verifier verdict, recovery analysis, or free-form chain-of-thought is generated.

4 Experiments and Results

4.1 Experimental setup

4.1.1 Implementation and training configuration

We use the post-trained Qwen3.5-9B multimodal model (Qwen Team, 2026) as the common base. Action, Contract, and Verifier SFT use rank-16 LoRA for two epochs; their learning rates are 2×10^{-5}, 10^{-6}, and 10^{-6}, respectively. GRPO uses rank 64, four epochs, learning rate 10^{-6}, $G = 5$, KL coefficient 0.01, clipping radius 0.2, and reward weights $(0.10, 0.70, 0.05, 0.05, 0.05, 0.05)$ for Action Type, Action Argument, State Delta, Recovery, Progress/Loop, and Online Environment. The vision encoder is frozen throughout. Formal distributed training uses seven 8-GPU nodes (56 A100 80GB GPUs) with EFA; evaluation uses 15 8-GPU nodes and one tensor-parallel-size-1 vLLM instance per GPU. External checkpoints are evaluated with their released prompt, parser, coordinate convention, and native no-thinking/direct-action path. We report full sample sets, reject runs with missing samples or API errors, and retain raw predictions and per-task records.

4.1.2 BENCHMARKS AND METRICS

We evaluate four offline and two executable benchmarks. **AndroidControl** contains diverse demonstrations from 833 Android applications and provides high- and low-level instructions; we report action accuracy on 9,980 test steps per split (Li et al., 2024). **GUI-Odyssey** targets cross-application mobile navigation; its evaluation contains 83,885 steps over random, task, device, and app splits, and we report overall step success rate (SR), type match (TM), and grounding rate (GR) (Lu et al., 2025a). **ScreenSpot** contains 1,272 grounding queries across mobile, desktop, and web interfaces (Cheng et al., 2024), while **ScreenSpot-Pro** contains 1,581 queries from high-resolution professional applications (Li et al., 2025a); both use target-hit accuracy. **AndroidWorld** provides 116 reproducible task templates across 20 Android applications (Rawles et al., 2025); we execute three instances per template and report episode success rate (Episode SR).

4.1.3 COMPARED MODELS

Qwen3.5-9B is a natively multimodal 9B foundation model with a hybrid linear/full-attention language backbone and serves as the unadapted base control (Qwen Team, 2026). We additionally compare seven released 7B–8B GUI agents. **Aguvis-7B-720P** is a cross-platform pure-vision agent trained in two stages for grounding and planning (Xu et al., 2025b). **GUI-R1-7B** applies GRPO with unified action-rule rewards to a compact curated set (Luo et al., 2025). **ScaleCUA-7B** scales cross-platform computer-use data over six operating systems (Liu et al., 2026c). **GUI-Libra-8B** combines action-aware supervision with KL-regularized partially verifiable RL (Yang et al., 2026). **OpenMobile-8B** synthesizes mobile tasks and trajectories through exploration and policy switching (Cheng et al., 2026). **MemGUI-8B-SFT** uses Context-as-Action to proactively fold long-horizon interaction history (Liu et al., 2026a). **ForgeQwen3-8B** is the Qwen3-VL checkpoint adapted by MobileForge using automatically generated target-app tasks and hierarchical trajectory/step feedback (Liu et al., 2026b).

4.2 OFFLINE BENCHMARK RESULTS

Under the action-only protocol, with parse failures treated as errors, Table 1 indicates that TRAVEC-9B improves joint action selection and grounding rather than optimizing either component in isolation. Native Qwen3.5-9B ranks strongly on GUI-Odyssey grounding but not on type matching or step success, showing that localization alone does not yield executable decisions. TRAVEC-9B preserves this grounding strength while improving the other two metrics, indicating tighter coupling among perception, action selection, and transition progress. ForgeQwen3-8B slightly leads in type matching, yet its weaker grounding and step success show that a correct action class remains insufficient without accurate arguments and trajectory-aware execution.

The advantage widens on AndroidControl Low, suggesting that TRAVEC-9B makes effective use of explicit action-level instructions. The smaller High margin indicates that higher-level task decomposition remains more challenging. Its ScreenSpot-Pro result extends this pattern to visually dense professional interfaces and shows transfer beyond mobile trajectories. TRAVEC-9B also retains the compact generation behavior of the base model while reducing latency and completion length. Specialized GUI baselines require slower and longer requests without matching its overall accuracy. This joint pattern supports moving verification and recovery reasoning into training while keeping the deployed policy action-only. Because Table 1 contains offline metrics, the evidence establishes per-action capability and efficiency rather than complete task execution.

4.3 ONLINE BENCHMARK RESULTS

Table 2 evaluates whether the offline gains transfer to executable task completion under the benchmark-specific success protocols.

TRAVEC-9B ranks first on AndroidWorld and improves over its Qwen3.5-9B base by 14.655 percentage points. Its margin over the strongest specialized baseline is only 0.862 points, showing that the improvement is substantial relative to initialization but narrow at the frontier. Thus the improvement is not a uniform scaling effect across executable environments.

Table 2: Online benchmark performance on AndroidWorld, reported as episode success rate. Best and second-best results are bolded and underlined.

Model	AndroidWorld ↑
Aguvis-7B-720P	9.483
GUI-R1-7B	12.931
ScaleCUA-7B	14.655
GUI-Libra-8B	19.828
OpenMobile-8B	18.103
MemGUI-8B-SFT	25.000
ForgeQwen3-8B	<u>34.483</u>
Qwen3.5-9B	20.690
TRAVEC-9B (ours)	**35.345**

The rank reversal separates control in the rollout domain from long horizon cross application orchestration. AndroidWorld matches the resettable rollout environment. TRAVEC-9B instead compresses auxiliary supervision into its deployed policy. This design works within the rollout domain, but broader experience and task memory remain important for transfer.

4.4 Ablating process-audit and supervision components

Table 3 isolates how process auditing and auxiliary supervision shape the rollout data used by the Action Policy.

Removing Action SFT weakens every evaluation regime, showing that audited GRPO depends on the action and interface prior learned during SFT. Other omissions reorder the benchmarks, indicating complementary components whose utility varies by task. Difficult grounding is most sensitive to structured contract fields, while static metrics expose little benefit from recovery supervision. Retaining more steps without Critic filtering offers no consistent gain, so the Critic improves rollout quality rather than data volume. Since the formal TRAVEC-9B reference uses a different schedule, it is not a matched causal baseline.

Table 3: Component ablations. "w/o" denotes *without*. Best and second-best results are bolded and underlined.

Model	AndroidControl		ScreenSpot	
	High ↑	Low ↑	SS ↑	SS-Pro ↑
w/o Critic	58.048	82.097	92.452	63.061
w/o Contract	57.024	80.110	<u>92.610</u>	63.820
w/o Verifier	<u>59.573</u>	82.994	92.374	63.947
w/o Contract Fields	56.344	82.816	91.824	62.998
w/o Recovery	56.723	<u>86.553</u>	92.296	<u>64.643</u>
w/o Action SFT	49.453	74.170	89.937	56.610
TRAVEC-9B	**64.309**	**93.337**	**92.689**	**67.805**

4.5 Ablating process-reward weight

Table 4 reveals a nonmonotonic balance between action fidelity and process shaping. Mixed objectives outperform the action only and all process settings on most metrics, showing that transition evidence complements rather than replaces direct action supervision. Changing process mass also shifts the preferred setting between instruction following and visual grounding, so more process supervision is not uniformly better. The formal TRAVEC-9B vector follows this pattern by keeping action correctness dominant and process rewards bounded. Its broad advantage is consistent with this balance, but the schedule mismatch and absence of a matched 20% run prevent a claim that the deployed vector is optimal.

Table 4: Reward-weight ablation. Weights follow the order Type, Argument, State Delta, Recovery, Progress/Loop, and Environment. The separated final row denotes the TRAVEC-9B configuration. Best and second-best results are bolded and underlined.

Reward weights	AndroidControl		ScreenSpot	
	High ↑	Low ↑	SS ↑	Pro ↑
$(0.125, 0.875, 0, 0, 0, 0)$	54.890	78.816	92.060	62.998
$(0.0625, 0.4375, 0.125, 0.125, 0.125, 0.125)$	<u>56.523</u>	82.425	92.453	<u>63.820</u>
$(0.025, 0.175, 0.2, 0.2, 0.2, 0.2)$	55.631	<u>82.745</u>	**92.846**	63.631
$(0, 0, 0.25, 0.25, 0.25, 0.25)$	55.952	81.152	92.060	63.188
$(0.10, 0.70, 0.05, 0.05, 0.05, 0.05)$	**64.309**	**93.337**	<u>92.689</u>	**67.805**

5 Conclusion

We presented TRAVEC, a framework that turns transition prediction, evidence-based verification, and failure recovery into scalable annotation and reinforcement-learning signals for mobile GUI agents. Its automated pipeline builds 550K verifiable contract, negative-recovery, preference, and calibration annotations; two auxiliary adapters and a trajectory Critic audit resettable self-generated experience; and six rewards align the final policy with action correctness, observed state change, recovery, progress, loop avoidance, and execution outcome. The resulting TRAVEC-9B removes all auxiliary modules at deployment and emits only executable actions. Across the completed experiments, it improves substantially over its Qwen3.5-9B base and Action SFT on AndroidWorld while remaining strong on offline action and grounding benchmarks; the component and reward-weight ablations identify task-dependent trade-offs, while the matched efficiency test shows lower latency and shorter completions than the base model. More broadly, TRAVEC offers a route to move expensive reasoning and verification away from the online critical path and into auditable policy learning.

References

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Wang. Agent s: An open agentic framework that uses computers like a human. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/394c7c30ea87b5c3521b4d9e9d419071-Abstract-Conference.html.

Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s2: A compositional generalist-specialist framework for computer use agents, 2025b. URL https://arxiv.org/abs/2504.00906.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332, 2024. doi: 10.18653/v1/2024.acl-long.505. URL https://aclanthology.org/2024.acl-long.505/.

Kanzhi Cheng, Zehao Li, Zheng Ma, Nuo Chen, Jialin Cao, Qiushi Sun, Zichen Ding, Fangzhi Xu, Hang Yan, Jiajun Chen, Anh Tuan Luu, Jianbing Zhang, Lewei Lu, and Dahua Lin. Openmobile: Building open mobile agents with task and trajectory synthesis. In *Conference on Language Modeling (COLM)*, 2026. URL https://njucckevin.github.io/openmobile/.

Zihan Dong, Rui Qian, Qishi Zhan, Dongshen Peng, Kaixin Li, and Yu Li. Why are gui agents correct but late? decode on the decision-time critical path, tested with pre-compiled policy trees, 2026. URL https://arxiv.org/abs/2607.28399.

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

2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/4ca0e369689dadb25a5345ba9755ad6f-Abstract-Conference.html.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14281–14290, 2024. doi: 10.1109/CVPR52733.2024.01354. URL https://doi.org/10.1109/CVPR52733.2024.01354.

Kung-Hsiang Huang, Haoyi Qiu, Yutong Dai, Caiming Xiong, and Chien-Sheng Wu. Gui-kv: Efficient gui agents via kv cache with spatio-temporal awareness, 2025. URL https://arxiv.org/abs/2510.00536.

Runxi Huang, Liyu Zhang, Shengzhong Liu, and Xiaomin Ouyang. Mobileexplorer: Accelerating on-device inference for mobile gui agents via online exploration, 2026. URL https://arxiv.org/abs/2605.26546.

Hanyu Lai, Xiao Liu, Yanxiao Zhao, Han Xu, Hanchen Zhang, Bohao Jing, Yanyu Ren, Shuntian Yao, Yuxiao Dong, and Jie Tang. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007435.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use, 2025a. URL https://arxiv.org/abs/2504.07981.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2925. URL https://arxiv.org/abs/2406.03679.

Yanda Li, Chi Zhang, Wenjia Jiang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Advanced agent for flexible mobile interactions, 2025b. URL https://arxiv.org/abs/2408.11824.

Zehao Li, Zhenyu Wu, Yibo Zhao, Bowen Yang, Jingjing Xie, Zhaoyang Liu, Zhoumianze Liu, Kaiming Jin, Jianze Liang, Zonglin Li, Feng Wu, Bowen Zhou, Zun Wang, and Zichen Ding. Osthemis: A scalable critic framework for generalist gui rewards, 2026. URL https://arxiv.org/abs/2603.19191.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent, 2024. URL https://arxiv.org/abs/2411.17465.

Guangyi Liu, Gao Wu, Congxiao Liu, Pengxiang Zhao, Liang Liu, Mading Li, Zhang Qi, Mengyan Wang, Liang Guo, and Yong Liu. Memgui-agent: An end-to-end long-horizon mobile gui agent with proactive context management, 2026a. URL https://arxiv.org/abs/2606.19926.

Guangyi Liu, Pengxiang Zhao, Gao Wu, Yiwen Yin, Mading Li, Liang Liu, Congxiao Liu, Zhang Qi, Mengyan Wang, Liang Guo, Jiangning Zhang, and Yong Liu. Mobileforge: Annotation-free adaptation for mobile gui agents with hierarchical feedback-guided policy optimization, 2026b. URL https://arxiv.org/abs/2606.19930.

Zhaoyang Liu, JingJing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, Xuehui Wang, Qiushi Sun, Shi Liu, Weiyun Wang, Shenglong Ye, Qingyun Li, Zeyue Tian, Gen Luo, Xiangyu Yue, Biqing Qi, Kai Chen, Bowen Zhou, Yu Qiao, Qifeng Chen, and Wenhai Wang. Scalecua: Scaling open-source computer use agents with cross-platform data. In *International Conference on Learning Representations*, 2026c. URL https://iclr.cc/virtual/2026/poster/10006583.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices, 2025a. URL https://arxiv.org/abs/2406.08451.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent, 2024. URL https://arxiv.org/abs/2408.00203.

Zhengxi Lu, Jiabo Ye, Fei Tang, Yongliang Shen, Haiyang Xu, Ziwei Zheng, Weiming Lu, Ming Yan, Fei Huang, Jun Xiao, and Yueting Zhuang. Ui-s1: Advancing gui automation via semi-online reinforcement learning, 2025b. URL https://arxiv.org/abs/2509.11543.

Zhengxi Lu, Fei Tang, Guangyi Liu, Kaitao Song, Xu Tan, Jin Ma, Wenqi Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Ui-copilot: Advancing long-horizon gui automation via tool-integrated policy optimization, 2026. URL https://arxiv.org/abs/2604.13822.

Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents, 2025. URL https://arxiv.org/abs/2504.10458.

Jian Mu, Chaoyun Zhang, Chiming Ni, Lu Wang, Bo Qiao, Kartik Mathur, Qianhui Wu, Yuhang Xie, Xiaojun Ma, Mengyu Zhou, Si Qin, Liqun Li, Yu Kang, Minghua Ma, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. GUI-360°: A comprehensive dataset and benchmark for computer-using agents, 2025. URL https://arxiv.org/abs/2511.04307.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. Ui-tars: Pioneering automated gui interaction with native agents, 2025. URL https://arxiv.org/abs/2501.12326.

Qwen Team. Qwen3.5-9b. Hugging Face model repository, 2026. URL https://huggingface.co/Qwen/Qwen3.5-9B. Accessed 2026-08-26.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/01a83bc2f2732a58e6aa731e659e7101-Abstract-Conference.html.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. Os-genesis: Automating gui agent trajectory construction via reverse task synthesis, 2025. URL https://arxiv.org/abs/2412.19723.

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

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, Wanjun Zhong, Yining Ye, Yujia Qin, Yuwen Xiong, Yuxin Song, Zhiyong Wu, Aoyan Li, Bo Li, Chen Dun, Chong Liu, Daoguang Zan, Fuxing Leng, Hanbin Wang, Hao Yu, Haobin Chen, Hongyi Guo, Jing Su, Jingjia Huang, Kai Shen, Kaiyu Shi, Lin Yan, Peiyao Zhao, Pengfei Liu, Qinghao Ye, Renjie Zheng, Shulin Xin, Wayne Xin Zhao, Wen Heng, Wenhao Huang, Wenqian Wang, Xiaobo Qin, Yi Lin, Youbin Wu, Zehui Chen, Zihao Wang, Baoquan Zhong, Xinchun Zhang, Xujing Li, Yuanfan Li, Zhongkai Zhao, Chengquan Jiang, Faming Wu, Haotian Zhou, Jinlin Pang, Li Han, Qi Liu, Qianli Ma, Siyao Liu, Songhua Cai, Wenqi Fu, Xin Liu, Yaohui Wang, Zhi Zhang, Bo Zhou, Guoliang Li,

Jiajun Shi, Jiale Yang, Jie Tang, Li Li, Qihua Han, Taoran Lu, Woyu Lin, Xiaokang Tong, Xinyao Li, Yichi Zhang, Yu Miao, Zhengxuan Jiang, Zili Li, Ziyuan Zhao, Chenxin Li, Dehua Ma, Feng Lin, Ge Zhang, Haihua Yang, Hangyu Guo, Hongda Zhu, Jiaheng Liu, Junda Du, Kai Cai, Kuanye Li, Lichen Yuan, Meilan Han, Minchao Wang, Shuyue Guo, Tianhao Cheng, Xiaobo Ma, Xiaojun Xiao, Xiaolong Huang, Xinjie Chen, Yidi Du, Yilin Chen, Yiwen Wang, Zhaojian Li, Zhenzhu Yang, Zhiyuan Zeng, Chaolin Jin, Chen Li, Hao Chen, Haoli Chen, Jian Chen, Qinghao Zhao, and Guang Shi. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning, 2025a. URL https://arxiv.org/abs/2509.02544.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Zhang Ji, 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. doi: 10.52202/079017-0088. URL https://doi.org/10.52202/079017-0088.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, Bin Wang, Chuhan Wu, Yasheng Wang, Ruiming Tang, and Jianye Hao. Gui agents with foundation models: A comprehensive survey, 2025b. URL https://arxiv.org/abs/2411.04890.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: A foundation action model for generalist gui agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0faa4bc5f522076947a030273629d4fe-Abstract-Conference.html.

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

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1650. URL https://arxiv.org/abs/2404.07972.

Haiyang Xu, Xi Zhang, Haowei Liu, Junyang Wang, Zhaozai Zhu, Shengjie Zhou, Xuhao Hu, Feiyu Gao, Junjie Cao, Zihua Wang, Zhiyuan Chen, Jitong Liao, Qi Zheng, Jiahui Zeng, Ze Xu, Shuai Bai, Junyang Lin, Jingren Zhou, and Ming Yan. Mobile-agent-v3.5: Multi-platform fundamental gui agents, 2026. URL https://arxiv.org/abs/2602.16855.

Yifan Xu, Xiao Liu, Xinghan Liu, Jiaqi Fu, Hanchen Zhang, Bohao Jing, Shudan Zhang, Yuting Wang, Wenyi Zhao, and Yuxiao Dong. Mobilerl: Online agentic reinforcement learning for mobile gui agents, 2025a. URL https://arxiv.org/abs/2509.18119.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69772–69805. PMLR, 2025b. URL https://proceedings.mlr.press/v267/xu25ae.html.

Rui Yang, Qianhui Wu, Zhaoyang Wang, Hanyang Chen, Ke Yang, Hao Cheng, Huaxiu Yao, Baolin Peng, Huan Zhang, Jianfeng Gao, and Tong Zhang. Gui-libra: Training native gui agents to reason and act with action-aware supervision and partially verifiable rl, 2026. URL https://arxiv.org/abs/2602.22190.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, Jitong Liao, Qi Zheng, Fei Huang, Jingren Zhou, and Ming Yan. Mobile-agent-v3: Fundamental agents for gui automation, 2025. URL https://arxiv.org/abs/2508.15144.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms, 2024. URL https://arxiv.org/abs/2404.05719.

Shengcheng Yu, Yuchen Ling, Junyang Xing, Quan Zhou, Chunrong Fang, and Zhenyu Chen. Software engineering for and with gui agent, 2026. URL https://arxiv.org/abs/2608.09278.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, Jie Xie, Wei Zhou, Wang Xu, Yuanheng Zhang, Zhou Su, Zhongwu Zhai, Xiaoming Liu, Yudong Mei, Jianming Xu, Hongyan Tian, Chongyi Wang, Chi Chen, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2506.01391.

Hanzhang Zhou, Panrong Tong, Xu Zhang, Quyu Kong, Chenglin Cai, Tianyu Xia, Gongjie Zhang, Jianan Zhang, Long Li, Long Chen, Lei Wang, Gaole Dai, Pengxiang Li, Liangyu Chen, Yue Wang, and Steven Hoi. Qwen-ui-agent technical report: Toward next-generation real-world centric foundation gui agents, 2026a. URL https://arxiv.org/abs/2607.28227.

Xurui Zhou, Gongwei Chen, Yuquan Xie, Zaijing Li, Kaiwen Zhou, Shuai Wang, Shuo Yang, Zhuotao Tian, and Rui Shao. Hiconagent: History context-aware policy optimization for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Zhou_HiconAgent_History_Context-aware_Policy_Optimization_for_GUI_Agents_CVPR_2026_paper.pdf.