SCA: SPATIAL CREDIT ASSIGNMENT FOR REINFORCEMENT LEARNING OF GUI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

GUI agents automate tasks on digital devices by grounding language instructions in visual interfaces. Existing group-relative reinforcement learning improves GUI action prediction by comparing the rewards of multiple responses sampled from the same GUI state. However, binary evaluation treats spatially different failed clicks as identical and provides no relative signal when all sampled clicks fail. To address these limitations, we propose Spatial Credit Assignment (SCA), which uses the screen coordinates of sampled clicks to refine group-relative credit. Specifically, SCA predicts each held-out response's reward from the other responses in groups containing both successes and failures, then uses the prediction residual to adjust credit. When all sampled clicks fail, SCA instead orders them by distance to the annotated target. These spatial references are used only to construct the training update; the deployed policy remains unchanged. We evaluate whether this correction improves the policy update itself by comparing its error and directional alignment with the exact return gradient in a controlled synthetic study. Across GUI grounding and offline action-prediction benchmarks, SCA improves grounding across professional domains and achieves the strongest results among reinforcement-fine-tuned models on most action-prediction metrics, with consistent gains across the reported GUI suites.

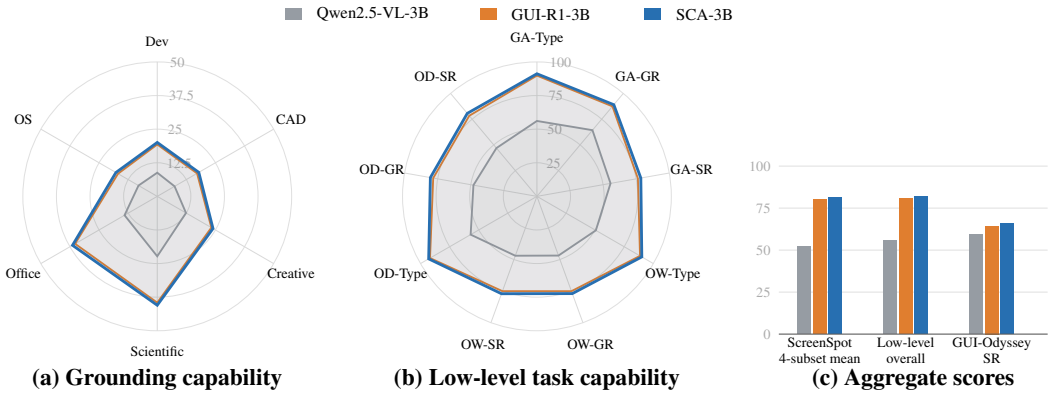
1 INTRODUCTION

GUI agents map screenshots and natural-language instructions to executable actions, such as clicking controls, selecting menu items, and entering text (Cheng et al., 2024; Luo et al., 2025; Lu et al., 2026; Wu et al., 2025; Gou et al., 2025). These tasks combine visual grounding, action-type selection, and argument generation across mobile, desktop, and web interfaces, so a useful policy must connect local screen geometry to the instruction rather than merely recognize an element. Recent work shows that reinforcement fine-tuning (RFT) can improve GUI action prediction with substantially less supervision than large-scale imitation learning (Luo et al., 2025; Lu et al., 2026). In particular, group-relative objectives such as GRPO and RLOO sample several responses to the same state and compare their rewards (Shao et al., 2024; Guo et al., 2025; Kool et al., 2019; Ahmadian et al., 2024). This training paradigm is well suited to GUI tasks because many sampled actions can be checked by a deterministic evaluator, making group credit the direct interface between evaluation and the policy update for all sampled responses during training.

However, reward-only group credit ignores the spatial structure of GUI actions. A binary evaluator gives the same reward to failed clicks at different locations, even when one click is close to the target and another points away from it. This produces two related challenges. In a mixed-hit group, successful and failed clicks reveal a local coordinate–reward relation, yet all misses receive the same group-relative credit. In an all-miss group, the group advantage is zero even though sampled clicks can lie at different distances from the target; reward-only credit therefore discards useful spatial evidence when successful clicks are rare.

Against this background, we propose Spatial Credit Assignment (SCA) for grouped GUI reinforcement learning. Specifically, SCA-Residual fits a held-out coordinate–reward trend for groups with mixed outcomes and blends the resulting residual with the ordinary group advantage. Moreover, SCA-Prox ranks actions by target proximity in all-miss groups, while all-hit groups retain the ordinary group credit. The spatial references are fitted within the current group and affect training only; inference therefore uses the trained policy alone, with no added predictor.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Grounding capability **(b) Low-level task capability** **(c) Aggregate scores**

Figure 1: **Performance overview across GUI benchmarks.** Panel (a) reports ScreenSpot-Pro percentages on a fixed 0–50 scale, averaging the Text and Icon columns within each domain. Panel (b) reports the Type, GR, and SR percentages on a 0–100 scale. Panel (c) summarizes the arithmetic mean of the four ScreenSpot subcolumns in Table 1, plus the Low-level Overall and GUI-Odyssey scores in Table 2. GA, OW, and OD denote GUI-Act-Web, OmniAct-Web, and OmniAct-Desktop.

We evaluate SCA on professional GUI grounding, web and desktop action prediction, and recorded cross-app mobile states. Figure 1 summarizes Tables 1 and 2. Across the completed three-seed suite, SCA reaches 26.1 across the twelve ScreenSpot-Pro subcolumns, 80.5 ± 0.41 on OmniAct-Desktop grounding, and 66.0 ± 0.52 on GUI-Odyssey. It obtains the best listed reinforcement-fine-tuning results on ScreenSpot-Pro and in ten of eleven low-level metrics. The comparison uses three-seed SCA means and listed baseline scores, while a separate synthetic study tests the spatial update against an exact return gradient.

Taken together, these results support three contributions:

- First, we formulate spatial credit assignment for grouped GUI reinforcement learning and identify the distinct mixed-hit and all-miss failure regimes of reward-only credit.
- Second, we develop SCA-Residual and SCA-Prox to use coordinate–reward trends and target proximity within the appropriate reward regimes.
- Third, we validate SCA with an exact-gradient analysis and a three-seed evaluation across GUI grounding and offline action-prediction benchmarks.

2 Related Work

Group-relative reinforcement learning. GRPO, RLOO, REINFORCE-style variants, and related clipping and sampling extensions construct credit from sampled rewards (Shao et al., 2024; Guo et al., 2025; Kool et al., 2019; Ahmadian et al., 2024; Hu et al., 2025; Yu et al., 2025). Recent agentic learning systems address long-horizon planning, phase-aware specialization, progress-aware updates, setwise multi-agent credit, and offline tool use (Li et al., 2026; Yang et al., 2026c;a;d; Lyu et al., 2026). A separate line of work incorporates action dependence through learned critics, analytic control variates, or factorized baselines (Gu et al., 2017; Liu et al., 2018; Wu et al., 2018; Tucker et al., 2018). These approaches motivate using more than reward values when constructing credit. For GUI clicks, however, reward-only group comparisons leave spatially distinct actions tied whenever their binary outcomes agree. SCA addresses this limitation by fitting a coordinate–reward reference within each group. Its leave-one-out construction excludes the scored response's reward from the fit and gate, drawing on the sample-splitting principle (Chernozhukov et al., 2018); the resulting update remains action-conditioned because the reference is evaluated at the sampled coordinate.

Spatial rewards for GUI agents. Spatial supervision offers another way to distinguish GUI actions. GUI-G^2 uses Gaussian reward modeling for GUI grounding (Tang et al., 2026), while SE-GUI studies self-evolutionary reinforcement learning for GUI agents (Yuan et al., 2025). Related work

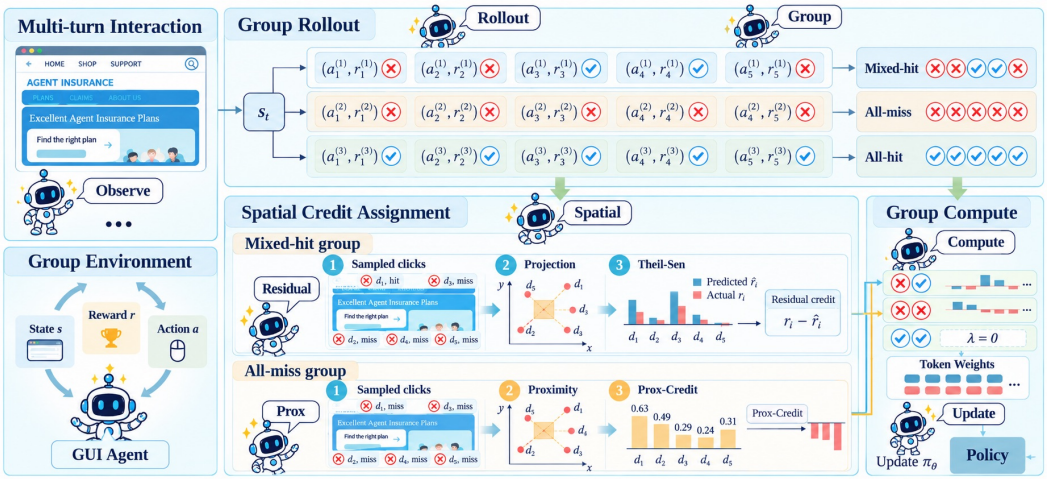


Figure 2: **Spatial credit within grouped GUI training.** At state s_t, the policy samples actions and observes their rewards. The upper panel shows group credit A_t^g and the spatial correction $A_t^S = A_t^F - A_t^g$. Below, Prox ranks all-miss clicks by proximity scores p_i, while Residual predicts mixed-hit rewards from held-out fits, forms residual credit, and blends it with group credit according to predictive skill. All-hit groups have zero spatial correction.

such as RSGround-R1 studies spatial reasoning in remote-sensing grounding (Huang et al., 2026). Target-distance shaping provides graded feedback even when a click misses the target, and SCA uses this signal in all-miss groups by standardizing and scaling target-proximity scores. Distance alone, however, assigns the same value to equally distant clicks and does not capture how reward varies with direction among the sampled actions. This motivates SCA's mixed-hit branch, which fits a coordinate–reward trend from the other responses and assigns credit from the held-out residual. SCA thus uses target proximity when binary outcomes are constant and the observed coordinate–reward relation when both hits and misses are available; Section 4 describes the corresponding dense-reward extension used in the comparison.

GUI grounding and navigation evaluation. Visual GUI agents such as SeeClick, OS-Atlas, ShowUI, and UGround connect language instructions to screen elements (Cheng et al., 2024; Wu et al., 2025; Lin et al., 2025; Gou et al., 2025). ScreenSpot and ScreenSpot-Pro evaluate grounding across interface types and professional domains, while GUI-Act, OmniAct, GUI-Odyssey, and AndroidControl cover action prediction and interaction settings (Cheng et al., 2024; Li et al., 2025; Chen et al., 2025; Kapoor et al., 2024; Lu et al., 2025; Li et al., 2024). Beyond GUIs, recent benchmarks study auto-bidding and agent-authored world models for sequential decision making (Yang et al., 2026b; Cai et al., 2026). Our evaluation focuses on fixed offline GUI states, including states drawn from trajectory datasets. In this setting, click coordinates and target annotations allow us to evaluate whether spatial credit improves grounding, while action-type and step-success metrics assess the accompanying changes in action prediction. Together, these benchmarks test whether spatial information improves GUI grounding and action prediction beyond reward-only group comparisons.

3 Preliminaries

SCA assigns credit after the sampled actions have been evaluated (Figure 2). In groups containing both successful and failed actions, Residual fits a coordinate–reward trend and uses deviations from that trend to refine the group advantage. In groups in which every sampled action fails, Prox assigns credit from distance to the annotated target. All-hit groups retain the standard group advantage. The spatial references are computed within each group and discarded after the update; inference therefore uses the trained policy alone during deployment.

Policy loss and credit weighting. Let M_{it} denote the response-token mask and $\rho_{it} = \pi_\theta(y_{it} \mid y_{i,<t}, x)/\pi_{\theta_{\text{old}}}(y_{it} \mid y_{i,<t}, x)$ the token likelihood ratio. The scalar credit A_i returned by SCA is detached and shared across the tokens of response i. We minimize the clipped actor loss

$$\mathcal{L}_{\text{actor}} = -\frac{\sum_{i,t} M_{it}\ell_{it}}{\sum_{i,t} M_{it}} + \lambda_{\text{KL}}\widehat{D}_{\text{KL}}, \tag{1}$$
$$\ell_{it} = \min\{\rho_{it}\,\text{sg}(A_i), \text{clip}(\rho_{it}, 1-\varepsilon_c, 1+\varepsilon_c)\,\text{sg}(A_i)\}\,.$$

Here sg denotes stop-gradient, $\varepsilon_c = 0.2$, and $\lambda_{\text{KL}} = 10^{-2}$. The KL term regularizes the policy against the reference model; its sampled estimator is specified in Appendix D. SCA supplies the response credit, while the token mask, likelihood ratio, clipping operation, and KL term follow the base actor objective used throughout training.

Reward construction. The primary evaluator returns a binary success indicator for each sampled action: a click receives one when it satisfies the annotated target condition and zero otherwise, while non-click actions are scored by the corresponding deterministic field checks. This reward is intentionally sparse, which exposes the tied-miss and all-miss cases addressed by SCA. For the dense-reward study, we additionally combine the Gaussian click reward and format-validity reward as $y_i = 0.8R_{\text{Gaussian},i} + 0.2R_{\text{format},i}$. The binary indicators still determine regime routing, whereas y_i supplies the response signal inside the mixed-hit residual fit. Thus the reward design provides a controlled comparison between standard binary credit and a denser response channel without changing the evaluator used at inference.

GUI-agent interaction model. A GUI agent receives a screenshot together with a natural-language instruction and emits a structured action. We represent an action as an action type, such as click, type, scroll, or navigation, together with arguments in the coordinate frame of the resized screenshot. For click actions, the argument is a point $a_i = (u_i, v_i)$; for text actions, it is a string field; and for scroll or navigation actions, it is a direction or target field. The policy therefore couples visual grounding with action-type and argument prediction rather than predicting an unconstrained token sequence.

States, targets, and evaluation. Each training example contains a fixed GUI state x, an instruction, and an annotated target region or target point. The evaluator parses the response into the same action schema and checks the relevant fields deterministically. A click is successful when its coordinate satisfies the target-region criterion; other actions are scored by their type, argument validity, and task-specific field checks. This evaluator produces a binary success signal for grouped reinforcement learning while preserving the coordinates needed by the spatial credit estimator.

Grouped rollout and grounding feedback. For one state, the policy samples N responses under the same instruction and screenshot. Their responses form a group on which relative credit is computed, so the comparison is across candidate actions for one state rather than across time steps of a trajectory. This distinction matters for GUI grounding: two responses can receive the same binary outcome while pointing to different screen locations, and an all-miss group can still contain a useful ordering by distance to the target. SCA uses this within-state spatial information only during training; the deployed agent still predicts the original action schema from the screenshot and instruction.

4 METHOD

For a GUI state x, the policy $\pi_\theta(a \mid x)$ samples N responses. A valid click has a coordinate $a_i \in \mathbb{R}^2$ and reward r_i. We write $z(v)_i = (v_i - \bar{v})/(s(v) + \epsilon)$ for group standardization, where $s(v)$ is the sample standard deviation. Binary GRPO uses $A_i^{\text{GRPO}} = z(r)_i$. All misses in a mixed group then receive the same advantage, and a group containing only misses has zero advantage. SCA uses click locations to distinguish these tied outcomes. Here, credit is assigned across responses to one state, not across the time steps of a trajectory. Each response's credit subsequently weights its token sequence.

All coordinates use the resized-image pixel frame shared by the image processor, annotations, and parsed clicks. After computing credit, we detach it and assign it to the response tokens in the clipped actor objective (Schulman et al., 2017). SCA variants use a clip ratio of 0.2 and KL coefficient 10^{-2}. Appendix D gives the token-level objective together with its pseudocode.

4.1 SPATIAL CREDIT ASSIGNMENT

Predicting reward from neighboring clicks. Residual uses successful and failed clicks in a mixed-hit group to estimate how reward varies with position. We project the coordinates onto four axes: horizontal, vertical, and the two diagonals. For each rollout i, it fits a line to the other $N-1$ coordinate–reward pairs on each axis and predicts the reward at a_i. We use the Theil–Sen estimator, whose slope is the median pairwise slope and whose intercept is the median residual (Theil, 1950; Sen, 1968). At the group size $N = 5$, each outer fit uses four points. The four projections capture directional variation without fitting a two-dimensional model to these few samples. Holding out the scored response prevents its reward from determining its own fitted reference value.

Selecting a predictive spatial trend. We select directions by their predictive accuracy, using a second leave-one-out loop within the outer training set. A direction contributes when its held-out predictions have positive correlation with reward and lower squared error than the fold-local reward mean. Its weight is proportional to that error reduction. The resulting weighted prediction, $\hat{r}_i$, gives a reference reward at the sampled location. We score the combined validation predictions in the same way to obtain q_i, then increase the spatial weight w_i linearly from zero at $q_i = 0.80$ to one at $q_i = 0.99$. This q band is the synthetic predictive-quality calibration in Appendix B. It is distinct from the runtime correlation blend thresholds $[\rho_{\mathrm{lo}}, \rho_{\mathrm{hi}}] = [0.3, 0.7]$ used by the training implementation; the two normalized scores are not interchangeable. The full nested calculation is given in Appendix C.1.

Assigning residual credit. The difference $r_i - \hat{r}_i$ measures how an action performs relative to the trend fitted from the other rollouts. For two misses, the action with the higher predicted reward has the more negative residual: it failed where the group suggested a better outcome. Figure 2 We standardize the residuals over the finite-prediction subset V to obtain $A_i^{\mathrm{res}} = z_V(r_V - \hat{r}_V)_i$. Residual blends them with the original group advantage and standardizes the blended vector:

$$A_i^{\mathrm{Residual}} = \left[z\big(w \odot A^{\mathrm{res}} + (1-w) \odot A^{\mathrm{GRPO}}\big)\right]_i, \tag{2}$$

where $\odot$ denotes elementwise multiplication. Figure 2 A miss can receive positive final credit because the update is relative to its group; Appendix C works through a five-click example. Invalid predictions have zero residual and weight. If all weights are zero, Residual returns A^{GRPO} directly. Proposition 1 describes the held-out reward separation, and Appendix C analyzes the resulting action-conditioned update.

Ordering all-miss clicks. An all-miss group has no reward variation from which to estimate a trend. Prox instead measures distance to the annotated target. For its centroid c_T and diagonal d_T, we use the length scale $\ell_T = \max(0.5\max(d_T, 1) + 50, \epsilon)$, with a 280-pixel diameter for point annotations. For valid clicks $i \in V_T$, proximity and credit are

$$p_i = \exp\left(-\frac{\|a_i - c_T\|_2}{\ell_T}\right), \tag{3}$$

$$A_i^{\mathrm{Prox}} = \alpha_T z_{V_T}(p_{V_T})_i, \tag{4}$$

with zero credit outside V_T. The coefficient $\alpha_T = 0.3\min(\mathrm{std}_0(p_{V_T})/0.15, 1)$ scales the branch by the spread of the proximity scores. Thus closer clicks receive higher credit, and a group of nearly equidistant misses receives a small update. Figure 2

SCA uses Prox for all-miss groups, Residual for groups containing both successful and failed actions, and the original group advantage for all-hit groups. With dense reward, routing still follows the binary success indicators h_i. The response channel $y_i = 0.8R_{\mathrm{Gaussian},i} + 0.2R_{\mathrm{format},i}$ supplies the Residual residual, while a parallel fit on h_i supplies its gate. The dense-reward variant uses $z(y)$ for the remaining groups. Appendix D specifies the update, including degenerate fits and invalid coordinates for each rollout.

Regime routing and edge cases. For each sampled state, the evaluator returns a binary reward vector and the group is assigned to exactly one regime. Mixed-hit groups use the held-out spatial trend because the group contains both positive and negative outcomes. groups in which every sampled action fails bypass regression because the response rewards are constant; Prox supplies the only directional signal available from the annotated target. All-hit groups retain the base group advantage because there is no failed response whose credit needs correction. Invalid coordinates, tied proximity scores, and failed fits are masked before standardization. If no valid spatial prediction remains, the

implementation returns the original group advantage, so the estimator has a defined fallback for every sampled rollout.

Cross-fitting and computational cost. The scored response is excluded from the fit used to predict its reference value. This prevents its own reward from entering the prediction and then being used to score the same response. The directional ensemble uses four one-dimensional projections and a second leave-one-out validation pass; with the fixed group size $N = 5$, each fit uses only four training points. Thus the additional work is confined to small regressions over the rollout group and does not introduce a learned critic or an inference-time network.

Relation to the policy update. After routing, SCA standardizes the selected credit and assigns one scalar to each sampled response. That scalar is detached before weighting response tokens in the clipped objective. The evaluator, sampler, clipping rule, and KL term therefore remain unchanged. This separation lets the experiments test the credit estimator itself: the main tables measure downstream behavior, while the synthetic gradient study measures whether the resulting update preserves the direction of the exact return gradient.

5 Experiments

5.1 Implementation Details

Training and inference. We initialize SCA from Qwen2.5-VL-3B-Instruct and use the EasyR1 implementation of the clipped PPO objective with five sampled responses per GUI state. All SCA seeds use the same data, prompt, evaluator, optimizer family, and rollout protocol. Appendix A.1 summarizes the training setup. Appendix F gives the prompt and parsed action fields. Tables 1 and 2 report means and sample standard deviations over three independently trained SCA seeds. Baseline rows contain the comparison point estimates reported by GUI-R1 (Luo et al., 2025) and UI-R1 (Lu et al., 2026). We report descriptive differences between these results.

Benchmarks and metrics. ScreenSpot and ScreenSpot-Pro measure click grounding, with ScreenSpot-Pro covering six professional interface domains. GUI-Act-Web, OmniAct-Web, and OmniAct-Desktop measure low-level action prediction across web and desktop interfaces. GUI-Odyssey supplies recorded states from cross-app mobile tasks. Following GUI-R1, we report action-type accuracy (Type), click-grounding accuracy (GR), and step success rate (SR). Appendix A.2 defines each endpoint and its evaluation protocol.

5.2 Experimental Results

Grounding capability. SCA scores above the listed GUI-R1 point estimates in all sixteen grounding subcolumns of Table 1, with differences of 0.3–1.7 percentage points. The arithmetic mean of the twelve ScreenSpot-Pro subcolumns is 26.1, compared with 25.2 for GUI-R1. On ScreenSpot, the SCA results are 90.5 ± 0.30 and 73.5 ± 0.42 for Web Text and Icon, and 94.8 ± 0.28 and 66.5 ± 0.45 for Desktop Text and Icon.

Cross-benchmark consistency. The gains are not restricted to a single evaluator or interface family. ScreenSpot-Pro tests professional grounding across six domains, while GUI-Act-Web and OmniAct combine action type, grounding, and step success. The same direction across these settings suggests that the update changes the quality of credit assigned during training rather than exploiting one benchmark-specific score during training.

Error regime and target type. The two spatial branches address different failure modes. In mixed-hit groups, the reward residual is useful because the group contains both positive and negative observations; in groups in which every sampled action fails, target proximity is the only available ordering signal. The text/icon split provides a complementary diagnostic: improvements on both target types indicate that the training signal is useful across visual target types, while the remaining icon gap identifies a perception bottleneck rather than a missing reward signal.

Mechanism check. The controlled synthetic study isolates credit assignment from VLM perception. We compare the estimated update with the exact return gradient using gradient MSE, cosine similarity,

Table 1: **GUI grounding results on ScreenSpot-Pro and ScreenSpot.** ScreenSpot-Pro contains six domains with text/icon subsets; ScreenSpot contains Web and Desktop text/icon subsets. SCA reports mean ± sample standard deviation over three independent training runs; other rows reproduce reported values. All RFT rows use a 3B-scale policy. Bold marks the highest RFT value.

	ScreenSpot-Pro												**ScreenSpot**			
Model	Dev		CAD		Creative		Scientific		Office		OS		Web		Desktop	
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon
Supervised fine-tuning																
SeeClick	0.6	0.0	2.5	0.0	1.0	0.0	3.5	0.0	1.1	0.0	2.8	0.0	55.7	32.5	72.2	30.0
OS-Atlas-4B	7.1	0.0	2.0	0.0	3.0	1.4	9.0	5.5	5.1	3.8	5.6	0.0	82.6	63.1	72.1	45.7
ShowUI-2B	16.9	1.4	2.5	0.0	9.1	0.0	13.2	7.3	15.3	7.5	10.3	2.2	–	–	–	–
CogAgent-18B	14.9	0.7	7.1	3.1	9.6	0.0	22.2	1.8	13.0	0.0	5.6	0.0	70.4	28.6	74.2	20.0
Aria-GUI	16.2	0.0	7.6	1.6	23.7	2.1	27.1	6.4	20.3	1.9	4.7	0.0	–	–	–	–
UGround-7B	26.6	2.1	14.2	1.6	27.3	2.8	31.9	2.7	31.6	11.3	17.8	0.0	80.4	70.4	82.5	63.6
Claude-3.5-Sonnet	22.0	3.9	14.5	3.7	25.9	3.4	33.9	15.8	30.1	16.3	11.0	4.5	–	–	–	–
OS-Atlas-7B	33.1	1.4	12.2	4.7	28.8	2.8	37.5	7.3	33.9	5.7	27.1	4.5	90.8	74.2	91.7	62.8
QwenVL2.5-3B	20.3	1.8	11.2	4.7	24.6	2.8	39.5	6.4	28.6	5.7	17.8	2.2	73.0	48.5	85.7	46.2
QwenVL2.5-7B	31.4	1.8	15.7	5.1	27.3	3.5	40.7	7.9	39.7	8.9	32.4	6.9	87.8	68.2	90.3	62.8
Zero-shot																
QwenVL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	–	–	–	–
GPT-4o	1.3	0.0	2.0	0.0	1.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	–	–	–	–
QwenVL2.5-3B	16.2	1.4	10.2	4.7	23.3	1.4	38.2	6.4	24.3	3.8	15.0	1.1	60.8	43.5	70.1	35.0
QwenVL2.5-7B	33.1	2.1	12.2	6.3	23.7	3.5	36.8	7.3	37.8	7.5	30.8	6.9	86.9	65.1	89.7	60.0
Reinforcement fine-tuning																
UI-R1-3B	22.7	4.1	11.2	6.3	27.3	3.5	43.4	11.8	32.2	11.3	13.1	4.5	85.2	73.3	90.2	59.3
GUI-R1-3B	33.8	4.8	26.4	7.8	40.9	5.6	61.8	17.3	53.6	17.0	28.1	5.6	89.6	72.1	93.8	64.8
SCA-3B (Ours)	**35.0** ±0.45	**5.1** ±0.32	**27.5** ±0.40	**8.2** ±0.36	**42.0** ±0.42	**5.9** ±0.31	**63.0** ±0.50	**18.0** ±0.38	**55.0** ±0.46	**17.8** ±0.35	**29.5** ±0.43	**6.0** ±0.34	**90.5** ±0.30	**73.5** ±0.42	**94.8** ±0.28	**66.5** ±0.45

Table 2: **Offline GUI action-prediction results.** Action-type accuracy (Type), click-grounding accuracy (GR), and step success rate (SR). SCA reports mean ± sample standard deviation over three independent training runs; comparison rows reproduce reported values. Bold marks the highest RFT value.

Model	GUI-Act-Web			OmniAct-Web			OmniAct-Desktop			Low-lvl	GUI-Odyssey
	Type	GR	SR	Type	GR	SR	Type	GR	SR	Overall	SR
Supervised fine-tuning											
OS-Atlas-4B	79.22	58.57	42.62	46.74	49.24	22.99	63.30	42.55	26.94	50.71	64.58
OS-Atlas-7B	86.95	75.61	57.02	85.63	69.35	59.15	90.24	62.87	56.73	70.07	73.00
QwenVL2.5-3B	76.95	66.34	61.69	66.24	56.91	53.02	77.62	62.54	63.76	65.79	62.03
QwenVL2.5-7B	87.66	84.77	79.89	81.62	73.45	73.39	86.23	80.17	79.80	80.09	84.00
Zero-shot											
GPT-4o	77.09	45.02	41.84	79.33	42.79	34.06	79.97	63.25	50.67	54.46	28.39
QwenVL2.5-3B	56.10	64.28	55.61	50.63	46.89	47.02	56.95	47.97	46.89	55.65	59.32
QwenVL2.5-7B	86.59	84.39	78.63	79.15	71.32	73.39	84.74	79.89	79.66	79.05	87.08
Reinforcement fine-tuning											
UI-R1-3B	75.89	79.43	67.31	75.42	61.35	61.33	73.41	64.12	63.98	70.85	**66.44**
GUI-R1-3B	89.86	87.42	76.31	88.58	75.10	75.08	91.86	78.37	78.31	80.88	64.41
SCA-3B (Ours)	**91.2** ±0.35	**89.0** ±0.42	**78.5** ±0.50	**90.1** ±0.38	**77.0** ±0.45	**77.1** ±0.47	**93.0** ±0.32	**80.5** ±0.41	**80.6** ±0.43	**82.0** ±0.36	66.0 ±0.52

and norm ratio. The results support the interpretation of the main tables: spatial credit reduces update error while preserving the direction of policy improvement, and the all-miss auxiliary is used only when the reward-based fit is degenerate.

Offline action prediction. Table 2 extends the comparison to action type, grounding, and step success. SCA reaches 89.0 ± 0.42 GUI-Act-Web GR, 77.0 ± 0.45 OmniAct-Web GR, and 80.5 ± 0.41 OmniAct-Desktop GR. These exceed the GUI-R1 entries by 1.58, 1.90, and 2.13 percentage points. The corresponding SCA step success rates are 78.5 ± 0.50, 77.1 ± 0.47, and 80.6 ± 0.43, and the reported low-level overall score is 82.0 ± 0.36.

On GUI-Odyssey, SCA reaches 66.0 ± 0.52, compared with 64.41 for GUI-R1 and 66.44 for UI-R1. This is 1.59 points above GUI-R1 and 0.44 below UI-R1. SCA leads the listed RFT point estimates in the ten low-level columns, while UI-R1 has the highest RFT value in the GUI-Odyssey column.

What drives the gain. The pattern across the two tables is consistent with the intended credit mechanism. The largest improvements appear in click grounding, whereas action-type accuracy changes less; this is expected because SCA changes the credit assigned to spatial actions and leaves the action parser and evaluator unchanged. The same ordering is visible across Web and Desktop settings, which reduces the chance that the gain comes from a single interface family.

Text versus icon targets. Text targets are consistently easier than icon targets for both GUI-R1 and SCA. SCA improves both groups, but the smaller absolute change on icons indicates that the method improves credit assignment without removing the underlying visual ambiguity of small or semantically weak targets. This distinction matters for interpretation: the method supplies a better training signal, while perception quality remains a separate bottleneck.

Regime-specific interpretation. The mixed-hit branch can refine the relative credit of misses using neighboring coordinate–reward pairs; the all-miss branch is the only source of a directional signal when binary rewards are constant. The exact-gradient study supports this separation: Residual remains aligned with the return gradient, while the full rule adds the Prox signal only in the regime where the residual fit is degenerate.

5.3 Analysis of the Main Results

Differences across professional domains. Table 3 summarizes the ScreenSpot-Pro results by domain. SCA scores higher in all six domains, with gains ranging from 0.70 to 1.10 percentage points. The largest increase is in Office, followed by Scientific, while the remaining domains show gains between 0.70 and 0.90 points. These differences show that the improvement extends across several types of professional interfaces.

Text and icon targets. Averaged over the six professional domains, SCA scores 42.00 on text targets and 10.17 on icon targets, compared with 40.77 and 9.68 for GUI-R1. The respective improvements are 1.23 and 0.48 points. Icon grounding remains substantially harder for both models, and the absolute gain is smaller on icons than on text. The gains are spread across the professional domains, while icon grounding remains the lower-scoring target category.

Table 3: **ScreenSpot-Pro domain means (%).** Text and icon scores are equally weighted; Δ denotes the gain over GUI-R1.

Domain	GUI-R1	SCA	Δ
Dev	19.30	20.05	+0.75
CAD	17.10	17.85	+0.75
Creative	23.25	23.95	+0.70
Scientific	39.55	40.50	+0.95
Office	35.30	36.40	+1.10
OS	16.85	17.75	+0.90

5.4 Exact-Gradient Analysis

The exact-gradient contextual-bandit analysis in Table 4 separates the credit estimator from VLM perception. Binary GRPO has gradient MSE 0.16201 on the four-context mixture. SCA-Residual reaches 0.14825 with cosine 0.99985 to the exact mean gradient, and Full SCA reaches 0.14754 with cosine 0.99997. The OLS-LOO control has lower MSE but a negative cosine and a norm ratio of 0.071: its mean update is small and points away from the return gradient. MSE measures estimation error, while cosine and norm ratio describe the expected update's direction and scale. Reading them together, Residual reduces MSE by about 8.5% relative to GRPO and retains positive alignment with the return gradient in this controlled mixture.

Why the exact-gradient check matters. This experiment is deliberately separated from benchmark accuracy. It keeps the sampled actions and return definition fixed, so changes in MSE reflect credit construction rather than a different policy or evaluator. The comparison also distinguishes three cases: a method can reduce error while changing direction, preserve direction while mis-scaling the update, or improve both. SCA's residual correction is useful only when its added term offsets estimation error; the exact-gradient reference lets us test that condition directly. Thus, the study connects the spatial rule to the quantity optimized during training and clarifies why matching a reward predictor alone would be insufficient.

Each estimator uses the same sampled actions. The target-distance LOO control in Appendix E has mean horizontal update -0.01560, compared with the exact value 0.08143. This control fits a residual reference against distance, rather than using distance as a reward. Its result highlights the importance of the reference's action dependence, which Appendix E.2 relates to gradient error.

Table 4: Exact-gradient mechanism analysis. Monte Carlo estimates on a fixed four-context mixture. Norm ratio is $\|\mathbb{E}\hat{g}\|_2/\|g_R\|_2$.

Estimator	Gradient MSE	Cosine	Norm ratio
Binary GRPO	0.16201	0.99995	1.624
2-D OLS-LOO	0.10082	-0.57269	0.071
SCA-RESIDUAL	0.14825	0.99985	1.524
Full SCA	0.14754	0.99997	1.597

The controlled perturbations in Appendix E.3 give a more detailed comparison of median and OLS fits at $N = 5$.

Interpreting gradient error. The exact reference also separates the error of the average update from variation across sampled groups. For an estimator $\hat{g}$ with mean $\bar{g} = \mathbb{E}\hat{g}$,

$$\mathbb{E}\|\hat{g} - g_R\|_2^2 = \mathbb{E}\|\hat{g} - \bar{g}\|_2^2 + \|\bar{g} - g_R\|_2^2.$$

A lower MSE can therefore reflect reduced dispersion, a mean closer to the reference, or a trade-off between these terms. The OLS-LOO result illustrates why the direction and magnitude of the mean must also be inspected: its small norm suppresses the update while its negative cosine indicates an opposing direction. By comparison, Residual and Full SCA retain positive alignment, although their norm ratios remain above one. Their lower MSE thus accompanies a useful update direction without implying an unbiased estimator. Table 9 in Appendix E reports the mean vectors and their Monte Carlo standard errors, complementing the aggregate comparison in Table 4.

When spatial corrections help. The paired identity in Equation 8 further explains what a spatial correction must achieve. Relative to the GRPO update, the change in MSE contains the correction's squared magnitude and its interaction with the original estimation error. Error decreases when this interaction offsets the added squared magnitude. Reward-prediction accuracy alone does not determine that balance, because each response's credit is multiplied by its policy score before the group update is formed. Holding out the scored reward avoids direct reuse of that observation, while the prediction still depends on the sampled coordinate. Evaluating the final, normalized update therefore tests the combined effect of the spatial fit, its gate, and credit standardization. This connects the estimator construction to the exact-gradient comparison under the same sampled actions.

6 LIMITATIONS

The evaluation covers one 3B backbone at $N = 5$ on fixed GUI states, with target annotations available for Prox. The benchmark results measure the combined rule; attributing its gains to Residual or Prox requires matched branch and distance-control experiments. Gate sensitivity, interactions with reward shaping, and scaling across model sizes remain open empirical questions. The gate calibration uses synthetic data. Online environments such as OSWorld, WebArena, and MiniWoB (Xie et al., 2024; Zhou et al., 2024; Shi et al., 2017) require recovery from actions that change subsequent observations, which the recorded-state evaluation does not test.

7 CONCLUSION

SCA uses neighboring rollout geometry to assign credit to GUI clicks. Residual handles mixed-hit groups with a cross-fitted residual, while Prox supplies a proximity signal for all-miss groups. In the reported offline evaluation, the three-seed means are above the listed GUI-R1 point estimates on the grounding and action-prediction suites. The synthetic study measures update direction and error. Branch controls, online interaction, and broader backbones are left for future evaluation.

AI USE STATEMENT

Generative AI tools assisted with language editing, LaTeX formatting.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *ACL*, 2024.

Guangfeng Cai, Kaibing Yang, Shuo He, Yu Li, Shengtian Yang, Jiaqi Lv, and Lei Feng. Beyond next-observation prediction: Agent-authored world modeling for sequential decision making. arXiv preprint arXiv:2606.25421, 2026.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. GUICourse: From general vision language model to versatile GUI agent. In *ACL*, 2025.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Boyu Gou, 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 *ICLR*, 2025.

Shixiang Gu, Timothy P. Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. Q-Prop: Sample-efficient policy gradient with an off-policy critic. In *ICLR*, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Liu, Z. T. Xie, Y. K. Huang, Y. Q. Zhang, W. Q. Li, Z. H. Gou, Z. Shao, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization. arXiv preprint arXiv:2501.03262, 2025.

Shiqi Huang, Shuting He, and Bihan Wen. RSGround-R1: Rethinking remote sensing visual grounding through spatial reasoning. arXiv preprint arXiv:2601.21634, 2026.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. OmniACT: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *ECCV*, 2024.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019.

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. In *ACM Multimedia*, 2025.

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 *NeurIPS*, 2024.

Yu Li, Guangfeng Cai, Shengtian Yang, Han Luo, Shuo Han, Xu He, Dong Li, and Lei Feng. Phgpo: Pheromone-guided policy optimization for long-horizon tool planning. arXiv preprint arXiv:2602.13691, 2026.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. ShowUI: One vision-language-action model for GUI visual agent. In *CVPR*, 2025.

Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-dependent control variates for policy optimization via Stein's identity. In *ICLR*, 2018.

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. In *ICCV*, 2025.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Pengxiang Zhao, Guangyi Liu, Guanjing Xiong, and Hongsheng Li. UI-R1: Enhancing efficient action prediction of GUI agents by reinforcement learning. In *AAAI*, 2026.

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. arXiv preprint arXiv:2504.10458, 2025.

Zhiyi Lyu, Yewen Li, Longtao Zheng, Shengtian Yang, Lang Feng, Lei Feng, Peng Jiang, Kun Gai, Qingpeng Cai, and Bo An. Agentbrew: Offline tool-use agent learning from raw real-world trajectories. arXiv preprint arXiv:2609.05837, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Pranab Kumar Sen. Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association*, 63(324):1379–1389, 1968.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of Bits: An open-domain platform for web-based agents. In *ICML*, 2017.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. GUI-G^2: Gaussian reward modeling for GUI grounding. In *AAAI*, 2026.

Henri Theil. A rank-invariant method of linear and polynomial regression analysis, Parts I–III. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A*, 53:386–392, 521–525, 1397–1412, 1950.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *ICML*, 2018.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham M. Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *ICLR*, 2018.

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 *ICLR*, 2025.

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 *NeurIPS*, 2024.

Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. Progress-conditioned group policy optimization for long-horizon agentic tasks. arXiv preprint arXiv:2607.22724, 2026a.

Shengtian Yang, Yewen Li, Peng Jiang, Zhiyi Lyu, Bo An, Peng Jiang, Qingpeng Cai, and Lei Feng. Platformbid: An auto-bidding benchmark from a unified advertising platform's perspective. In *KDD*, 2026b.

Shengtian Yang, Yu Li, Shuo He, Yewen Li, Qingpeng Cai, Peng Jiang, and Lei Feng. Phase-aware mixture of experts for agentic reinforcement learning. In *ICML*, 2026c.

Shengtian Yang, Ziyu Xiong, Yu Li, Yewen Li, Qingpeng Cai, and Lei Feng. Srpo: Setwise relative policy optimization for multi-agent llms. arXiv preprint arXiv:2609.08452, 2026d.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *NeurIPS*, 2025.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. SE-GUI: Enhancing visual grounding for GUI agents via self-evolutionary reinforcement learning. In *NeurIPS*, 2025.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.

A Experimental Details

A.1 Training configuration and comparison sources

All SCA headline runs use Qwen2.5-VL-3B-Instruct with five responses per GUI state, the same EasyR1 clipped objective, prompt, evaluator, and optimizer family across seeds. The runtime correlation blend uses $[\rho_{\text{lo}}, \rho_{\text{hi}}] = [0.3, 0.7]$; the separate synthetic predictive-quality calibration is described in Appendix B. The main-table SCA entries are three-seed summaries supplied with the manuscript; baseline entries are comparison point estimates reported by the cited works. Comparisons are descriptive because paired test-set uncertainty, matched training compute, and exact evaluator equivalence have not been established for every row. Sample standard deviations therefore describe SCA seed variation and do not claim statistical significance against published baselines.

A.2 Metrics and aggregation

Tables 1 and 2 define the performance values used throughout the paper. Type denotes action-type accuracy, GR denotes click-grounding accuracy, and SR denotes step success. The GUI-Odyssey column concerns predictions at recorded states, rather than closed-loop task completion.

For ScreenSpot-Pro, the mean displayed in the text is the unweighted mean of the twelve domain–UI-type scores in Table 1. For ScreenSpot, Figure 1 averages only the four displayed Web/Desktop Text/Icon scores. The Low-level Overall column is reproduced from Table 2. All improvements in the text are absolute percentage-point differences, calculated before rounding. The sample SD in each SCA cell measures variation across training seeds. Baseline rows are comparison point estimates; the reported differences are descriptive rather than paired significance tests.

A.3 Recorded-case evaluator

The case examples use a parsed action type and its applicable argument. For a grounding action at reference (x^*, y^*), the coordinate acceptance test is $((x - x^*)/W)^2 + ((y - y^*)/H)^2 < 0.14^2$. Text arguments use token-set F1 at least 0.5; other actions use type match. The examples preserve their original screenshot, instruction, predicted coordinate, and evaluator outcome. These tests specify the acceptance rule used for the illustrated cases.

B Gate Calibration

The soft gate is calibrated independently of GUI training and test metrics. The candidate grid is $q_{\text{lo}} \in \{0.70, 0.75, 0.80\}$ crossed with $q_{\text{hi}} \in \{0.95, 0.975, 0.99\}$. For each candidate, a paired simulation draws $10{,}000$ groups from each of 17 pre-specified scenarios: five Bernoulli nulls, clean linear-logit and box signals, and coordinate-, label-, and combined-noise variants. We calibrate the

Table 5: Selection and one-shot holdout study for the fixed $N = 5$ gate (10,000 groups per scenario; selection seed 20260807; holdout seed 20260808). "Null active" and "null strong" are simultaneous 95% upper bounds; "clean lift" and "clean mean weight" are simultaneous lower bounds.

Constraint summary	Required	Selection	Holdout
Maximum null active-rollout rate	≤ 0.1000	0.0801	0.0793
Maximum null rollout rate with $w \geq 0.9$	≤ 0.0200	0.0171	0.0155
Minimum clean paired activation lift	≥ 0.0100	0.0139	0.0159
Minimum clean mean gate weight	≥ 0.0100	0.0179	0.0189

operating point at $N = 5$ and retain $N \in \{8, 16\}$ as out-of-distribution stress settings. All rate denominators include every generated rollout, including predictor-invalid cases.

The selection rule imposes 14 one-sided constraints with Bonferroni family-wise error rate 0.05: for each Bernoulli null, the simultaneous 95% upper bound on gate activation is at most 0.10 and on gate weight at least 0.9 is at most 0.02; for each clean signal family, the simultaneous lower bounds on paired activation lift and mean gate weight are at least 0.01. For each generated signal group, the paired lift is its active-rollout fraction minus the expected fraction under exact enumeration of all binary label assignments that preserve that group's hit count and coordinates; the reported bound averages these group-paired differences. Among feasible bands, the rule maximizes q_{lo} and then q_{hi}, selecting $[0.80, 0.99]$ with seed 20260807. Seeds 20260804–20260806 were used while developing the synthetic scenarios and candidate grid. The selected band was then evaluated once, without reselection, on seed 20260808 and passed all blocking constraints (Table 5). This holdout changes the random draw within the same synthetic family, providing a repeatability check for the selected band.

C Cross-Fitting Properties and Worked Example

C.1 Directional prediction and nested selection

For rollout i, let $I_{-i} = \{1, \ldots, N\} \setminus \{i\}$. The four unit directions are $(1, 0)$, $(0, 1)$, $(1, 1)/\sqrt{2}$, and $(1, -1)/\sqrt{2}$. The projected coordinate is $u_{j,d} = e_d^\top a_j$. Fitting on I_{-i} gives $\hat{r}_{i,d} = f_{i,d}(u_{i,d})$. In the inner loop, a fit on $I_{-i} \setminus \{j\}$ gives $\tilde{r}_{i,j,d}$, with null prediction $m_{i,j} = (N-2)^{-1} \sum_{k \in I_{-i} \setminus \{j\}} r_k$. For finite predictions with positive prediction–reward correlation and nonzero null error, the skill is

$$s_{i,d} = \left[1 - \frac{\sum_{j \in I_{-i}} (r_j - \tilde{r}_{i,j,d})^2}{\sum_{j \in I_{-i}} (r_j - m_{i,j})^2} \right]_+, \tag{5}$$

and it is zero otherwise. Positive skills give weights $\beta_{i,d} = s_{i,d} / \sum_{d'} s_{i,d'}$. These weights combine both the outer predictions and the inner validation traces: $\hat{r}_i = \sum_d \beta_{i,d} \hat{r}_{i,d}$ and $\tilde{r}_{i,j} = \sum_d \beta_{i,d} \tilde{r}_{i,j,d}$. Applying the same skill calculation to the combined trace gives q_i. The gate is $w_i = \mathrm{clip}((q_i - 0.80)/0.19, 0, 1)$. Binary predictions are clipped to $[-1, 2]$ before scoring or residualization. If no candidate has positive skill, rollout i has no valid prediction and receives zero spatial weight. Appendix D.1 details the remaining numerical cases.

C.2 Properties and worked example

Proposition 1 (Held-out reward separation)**.** *Consider a mixed-hit binary group with complete coordinates. For rollout i, fix the sampled actions and rewards r_{-i}. The cross-fitted prediction $\hat{r}_i$, candidate weights $\beta_{i,d}$, ensemble skill q_i, and gate w_i are invariant to changes in r_i that preserve the mixed-hit route.*

Proof. The outer candidates use I_{-i}, and every nested candidate uses a subset of I_{-i}. Candidate skill, aggregation weights, the ensemble trace, and the gate are therefore functions of $a_{1:N}$ and r_{-i}. The held-out reward is used later to form residual and group-relative credit; it is absent from its own prediction and gate under the fixed-route condition. □

For the sampled token sequence Y_i, let $S_i = \nabla_\theta \log \pi_\theta(Y_i \mid x)$ and let $H_i = H_i(a_{1:N}, r_{-i}; x)$ denote its cross-fitted spatial prediction. The simplified residual update satisfies

$$\mathbb{E}\left[\frac{1}{N}\sum_i S_i(r_i - H_i)\right] = g_R - \mathbb{E}\left[\frac{1}{N}\sum_i S_i H_i\right], \tag{6}$$

where $g_R = \nabla_\theta \mathbb{E}[r(Y, x)]$. The second term is the spatial correction introduced by action-conditioned credit. Since H_i is evaluated at the sampled action, excluding r_i does not imply $\mathbb{E}[S_i H_i] = 0$. Section E measures the resulting direction, scale, and error in the synthetic setting.

Inputs to spatial credit. Table 6 distinguishes the information used by each rule. Residual's feature map is $e_d^\top a_j$. Given the sampled actions, validity masks, and supplied response and routing channels, its calculation does not read a target box, centroid, or diagonal. Those channels may already encode target supervision through the evaluator. Prox and distance-based rules additionally use target geometry to construct their spatial signal.

Table 6: **Information used to construct credit.** The task evaluator supplies scalar rewards; spatial rules use the additional quantities listed.

Rule	Spatial reference	Target geometry
Binary GRPO	None; normalize binary group rewards	In the evaluator
Distance shaping	Fixed function of action-to-target distance	Read explicitly
Distance LOO	Reward trend fitted against target distance	Read explicitly
SCA-Residual	Reward trend fitted against sampled coordinates	In the evaluator
SCA-Prox	Exponential proximity, scaled after normalization	Read explicitly

Illustrative mixed group. Table 7 works through five sampled actions. GRPO assigns the same credit to all three misses, whereas Residual uses their different fitted references. The first miss has positive final credit: its observed reward 0 exceeds the extrapolated reference -0.423. The reference is a clipped linear prediction, not a success probability. The example illustrates relative residual credit, including its ability to assign a miss more credit than a hit in the same group.

Table 7: **A deterministic five-click calculation.** Values are rounded to three decimals. A dash denotes an invalid prediction, with zero gate weight.

Click a_i	Reward r_i	GRPO	Prediction $\hat{r}_i$	Gate w_i	Residual
$(0, 0)$	0	-0.730	-0.423	1.000	0.770
$(1, 2)$	0	-0.730	0.309	0.190	-1.095
$(2, 1)$	0	-0.730	0.309	0.190	-1.095
$(4, 5)$	1	1.095	–	0.000	0.710
$(5, 4)$	1	1.095	–	0.000	0.710

D ALGORITHM

With response-token mask M_{it} and likelihood ratio $\rho_{it} = \pi_\theta(y_{it} \mid y_{i,<t}, x)/\pi_{\theta_{\text{old}}}(y_{it} \mid y_{i,<t}, x)$, the SCA actor uses

$$\mathcal{L}_{\text{actor}} = -\frac{\sum_{i,t} M_{it} \min\{\rho_{it}\,\text{sg}(A_i), \text{clip}(\rho_{it}, 1-\varepsilon_c, 1+\varepsilon_c)\,\text{sg}(A_i)\}}{\sum_{i,t} M_{it}} + \lambda_{\text{KL}} \widehat{D}_{\text{KL}}, \tag{7}$$

where $\varepsilon_c = 0.2$ and $\lambda_{\text{KL}} = 10^{-2}$. The implemented low-variance KL term is the masked mean of $\text{clip}(\exp(\delta_{it}) - \delta_{it} - 1, -10, 10)$ with $\delta_{it} = \log \pi_{\text{ref}} - \log \pi_\theta$.

D.1 NUMERICAL IMPLEMENTATION

Degenerate projected pairs are omitted from median-slope fitting. A direction with no finite candidate prediction receives zero skill. Binary outer and nested predictions are clipped to $[-1, 2]$ before

Algorithm 1 Strict SCA-RESIDUAL credit for one binary-reward group

Require: $G = \{(a_i, r_i)\}_{i=1}^{N}$, directions $\{e_d\}_{d=1}^{D}$, gate band $[q_{\mathrm{lo}}, q_{\mathrm{hi}}]$
$A^{\mathrm{GRPO}} \leftarrow z(r)$
if all rewards are equal or any rollout coordinate is invalid **then**
  return A^{GRPO}
end if
for $i = 1, \ldots, N$ **do**
  $I_{-i} \leftarrow \{1, \ldots, N\} \setminus \{i\}$
  for direction d **do**
    Fit $f_{i,d}$ on I_{-i} and predict $\hat{r}_{i,d}$
    for $j \in I_{-i}$ **do**
      Fit on $I_{-i} \setminus \{j\}$ and predict $\tilde{r}_{i,j,d}$
      $m_{i,j} \leftarrow \mathrm{mean}\{r_k : k \in I_{-i} \setminus \{j\}\}$
    end for
    Clip binary outer and nested predictions to $[-1, 2]$
    Compute signed positive skill $s_{i,d}$ using Eq. (5)
  end for
  if at least one candidate has positive skill **then**
    Skill-weight candidates to obtain $\hat{r}_i$ and nested predictions
    Score aggregate nested predictions to obtain q_i
    if $\hat{r}_i$ is finite and $q_i > 0$ **then**
      $w_i \leftarrow \mathrm{clip}((q_i - q_{\mathrm{lo}})/(q_{\mathrm{hi}} - q_{\mathrm{lo}}), 0, 1)$
    else
      Mark i prediction-invalid; set $\hat{r}_i \leftarrow \mathrm{NaN}$ and $w_i \leftarrow 0$
    end if
  else
    Mark i prediction-invalid; set $\hat{r}_i \leftarrow \mathrm{NaN}$ and $w_i \leftarrow 0$
  end if
end for
if $\max_i w_i = 0$ **then**
  return A^{GRPO}
end if
$V \leftarrow \{i : \hat{r}_i \text{ is finite}\}$
$A_i^{\mathrm{res}} \leftarrow z_V(r_V - \hat{r}_V)_i$ for $i \in V$; 0 otherwise
return $z(w \odot A^{\mathrm{res}} + (1 - w) \odot A^{\mathrm{GRPO}})$

residualization and gate scoring. A non-positive prediction–reward correlation, non-finite statistic, degenerate null SSE, or non-positive skill sets the candidate quality to zero. In the gated mixed-hit Residual path, if no candidate remains or any coordinate in the calibrated $N = 5$ group is invalid, the group returns ordinary group-relative credit. The all-miss Prox path instead operates on its nonempty valid-coordinate subset as defined in Eq. (4). All standardizations use a stabilized sample standard deviation and return zero on a degenerate vector. The training launcher sets the gate band to $[0.80, 0.99]$.

Fit count. For D directions and N responses, Residual performs DN outer fits and $DN(N-1)$ inner fits. With $D = 4$ and $N = 5$, this is 20 four-point fits and 80 three-point fits per mixed-hit group. Each outer fit considers at most six coordinate pairs and each inner fit at most three. This work uses the completed rollout group; it requires no additional policy generations and no fitted predictor at inference.

E EXACT-GRADIENT ANALYSIS

E.1 SYNTHETIC SETTING

The audit uses a diagonal Gaussian policy $a \mid x_s \sim \mathcal{N}(\mu, I_2)$ with shared mean parameter $\mu = (0, 0)$ and an axis-aligned target box $T_s = [c_{s,1} - h_{s,1}, c_{s,1} + h_{s,1}] \times [c_{s,2} - h_{s,2}, c_{s,2} + h_{s,2}]$. The objective is the exact binary return $J_s(\mu) = \Pr(a \in T_s)$. For general policy standard deviation σ, let $L_k = (c_{s,k} - h_{s,k} - \mu_k)/\sigma$, $U_k = (c_{s,k} + h_{s,k} - \mu_k)/\sigma$, $p_k = \Phi(U_k) - \Phi(L_k)$, and $d_k = [\phi(L_k) - \phi(U_k)]/\sigma$. Then $J_s = p_1 p_2$ and $\nabla_\mu J_s = (d_1 p_2, p_1 d_2)$, which is the exact reference used in Table 8.

Table 8: Exact-gradient contextual-bandit scenarios. c is target center, h is target half-extent, and $P(\text{hit})$ is analytic at $\mu = (0, 0)$.

Scenario	c_x	c_y	h_x	h_y	$P(\text{hit})$	$\nabla_x J$	$\nabla_y J$
Dense offset	0.35	-0.20	1.20	1.20	0.56418	0.12015	-0.06842
Balanced offset	0.45	0.25	0.85	0.70	0.28076	0.09896	0.05948
Sparse offset	0.90	0.55	0.55	0.45	0.08733	0.07110	0.04489
Rare/far	1.45	-0.80	0.45	0.35	0.02615	0.03550	-0.02009

Table 9: Estimated mean gradient and component-wise Monte Carlo standard error on the uniform contextual mixture. The exact row has no sampling error.

Estimator	Mean $g_x \pm$ MCSE	Mean $g_y \pm$ MCSE
Exact reference	0.08143	0.00396
Binary GRPO	0.13220 ± 0.00045	0.00770 ± 0.00044
2-D OLS-LOO	-0.00307 ± 0.00034	-0.00489 ± 0.00034
2-D Ridge-LOO	-0.00304 ± 0.00034	-0.00490 ± 0.00034
SCA-RESIDUAL	0.12396 ± 0.00043	0.00818 ± 0.00042
Target-distance LOO	-0.01560 ± 0.00045	-0.00416 ± 0.00045
Full SCA	0.12995 ± 0.00043	0.00738 ± 0.00042
SCA-RESIDUAL without final gate	0.09891 ± 0.00041	0.00909 ± 0.00040

Table 8 gives every context. The audit uses seed 20260803, $\sigma = 1$, $N = 5$, and $100{,}000$ independently sampled groups per context. All estimators receive the same sampled actions. The uniform contextual mixture pools $400{,}000$ group gradients and compares their mean with the arithmetic mean of the four exact context gradients. These synthetic mixture outcomes support the estimator analysis.

E.2 GRADIENT ERROR

For a sampled response Y_i, let $S_i = \nabla_\theta \log \pi_\theta(Y_i \mid x)$. Before PPO clipping, the on-policy group update is $\hat{g} = N^{-1} \sum_i S_i A_i$. On the same sampled group, Residual and GRPO therefore satisfy $\hat{g}_C = \hat{g}_G + X$, where $X = N^{-1} \sum_i S_i (A_i^{\text{Residual}} - A_i^{\text{GRPO}})$. Writing g_R for the exact return gradient and expanding the squared errors gives

$$\mathbb{E}\|\hat{g}_C - g_R\|_2^2 - \mathbb{E}\|\hat{g}_G - g_R\|_2^2 = \mathbb{E}\|X\|_2^2 + 2\mathbb{E}\langle \hat{g}_G - g_R, X \rangle. \tag{8}$$

Thus gradient MSE decreases when the correction opposes the baseline error strongly enough to offset its own squared magnitude. This condition concerns the score-weighted, normalized update, whereas reward fitting concerns $r_i - \hat{r}_i$. Section 5.4 measures the former directly against the exact gradient in the synthetic mixture.

E.3 CONTROLLED CORRUPTION

We directly perturb one reward or one coordinate in $50{,}000$ synthetic groups per scenario, while measuring the change on the other rollouts. Table 10 reports gradient drift and the change in exact-gradient MSE. The two estimators are close in reward-flip settings; OLS is stronger for the balanced coordinate spike, while the median ensemble has a slightly smaller MSE increase for the sparse coordinate spike. The preferred fit therefore depends on the perturbation in these small-group settings.

Table 10: Controlled corruption study; smaller is better for both metrics.

Context	Corruption	Estimator	Drift	10^3 MSE Δ
Balanced	reward flip	median ensemble	0.146	-0.36
Balanced	reward flip	OLS ensemble	0.145	-1.32
Balanced	coordinate spike	median ensemble	0.042	9.24
Balanced	coordinate spike	OLS ensemble	0.038	8.21
Sparse	reward flip	median ensemble	0.021	-2.42
Sparse	reward flip	OLS ensemble	0.022	-2.61
Sparse	coordinate spike	median ensemble	0.009	2.35
Sparse	coordinate spike	OLS ensemble	0.009	2.40

F PROMPT TEMPLATES AND RECORDED CASES

This appendix makes the evaluation interface concrete. The prompt follows the GUI-R1 instruction format and is shared across the grounding and action-prediction benchmarks; only the admissible action set changes with the benchmark. Each recorded case below is drawn from the released evaluation replay and keeps the original screenshot, instruction, coordinates, and evaluator outcome.

F.1 PROMPT TEMPLATES

The ScreenSpot and ScreenSpot-Pro runs use the following user message, where `{instruction}` is the benchmark instruction and `{history}` is the action history (`None` for a single-step case):

Grounding Prompt

```
You are RUN1-R1, a reasoning GUI Agent Assistant. In this UI
screenshot <image>, I want you to continue executing the command
'{instruction}', with the action history being '{history}'. Please
provide the action to perform (enumerate from ['click']), the point
where the cursor is moved to (integer) if a click is performed, and
any input text required to complete the action. Output the thinking
process in <think></think>tags, and the final answer in
<answer></answer>tags as follows: <think>...
</think><answer>[{'action': 'click', 'point': [x, y], 'input_text':
'no input text [default]'}]</answer>
```

OmniAct uses the same message skeleton with a benchmark-specific action set:

Unified Action Prompt

```
You are GUI-R1, a reasoning GUI Agent Assistant. In this UI
screenshot <image>, I want you to continue executing the command
'{instruction}', with the action history being '{history}'. Please
provide the action to perform (enumerate from {action_set}), the
point where the cursor is moved to (integer) if a click is performed,
and any input text required to complete the action. Output the
thinking process in <think></think>tags, and the final answer in
<answer></answer>tags as follows: <think>...
</think><answer>[{'action': enum[...], 'point': [x, y],
'input_text': 'no input text [default]'}]</answer>
```

Benchmark Action Vocabularies

Benchmark	Action vocabulary
ScreenSpot / ScreenSpot-Pro	`click`
OmniAct-Web	`press_tab, moveto, rightclick, press_enter, scroll, click, press_down, hotkey, press_space, doubleclick`
OmniAct-Desktop	`click, moveto, press_pgdn`

The grounding script uses the identifier 'RUN1-R1'; the OmniAct scripts use 'GUI-R1'. For non-coordinate actions, the parser uses the sentinel point `[-100,-100]`; pointer actions use the benchmark screenshot coordinate frame. SCA receives the parsed coordinates and scalar rewards through this message/evaluator interface.

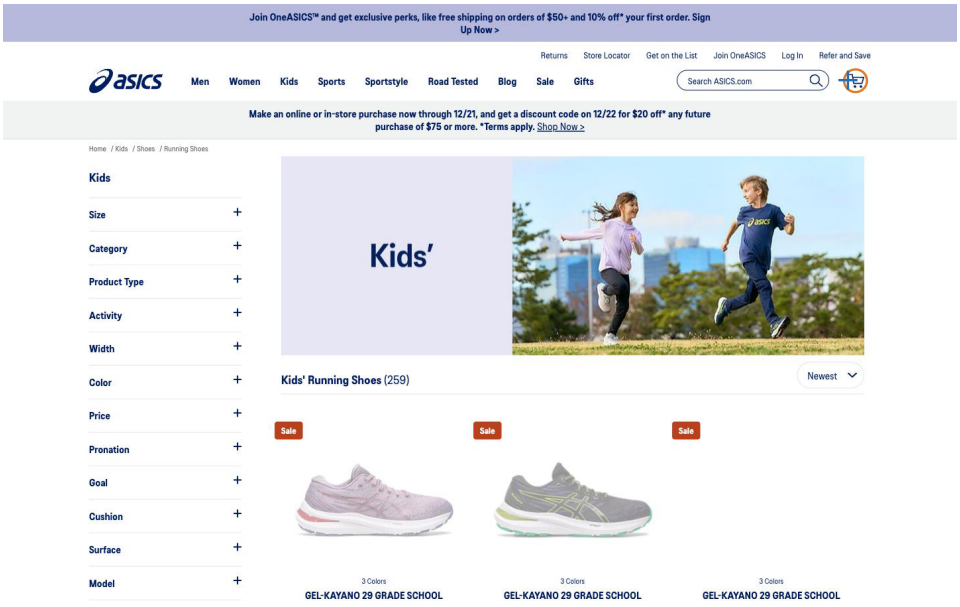
F.2 VERIFIABLE ACTION FIELDS

Each answer is parsed as $o_i = (o_i^{\text{act}}, o_i^{\text{point}}, o_i^{\text{text}})$. Evaluation checks the action type and its applicable argument: an accepted coordinate for pointer actions, token-set F1 ≥ 0.5 for text, and type match otherwise. We set $h_i = \mathbb{I}[o_i^{\text{act}} = g^{\text{act}}]\mathbb{I}[\text{ArgOK}(o_i, g)]$, with $h_i = 0$ for malformed outputs. Binary runs use $r_i = h_i$; shaped runs use h_i for SCA routing and the shaped scalar for policy credit.

F.3 RECORDED EVALUATION CASES

Case 1: Web Shopping Page

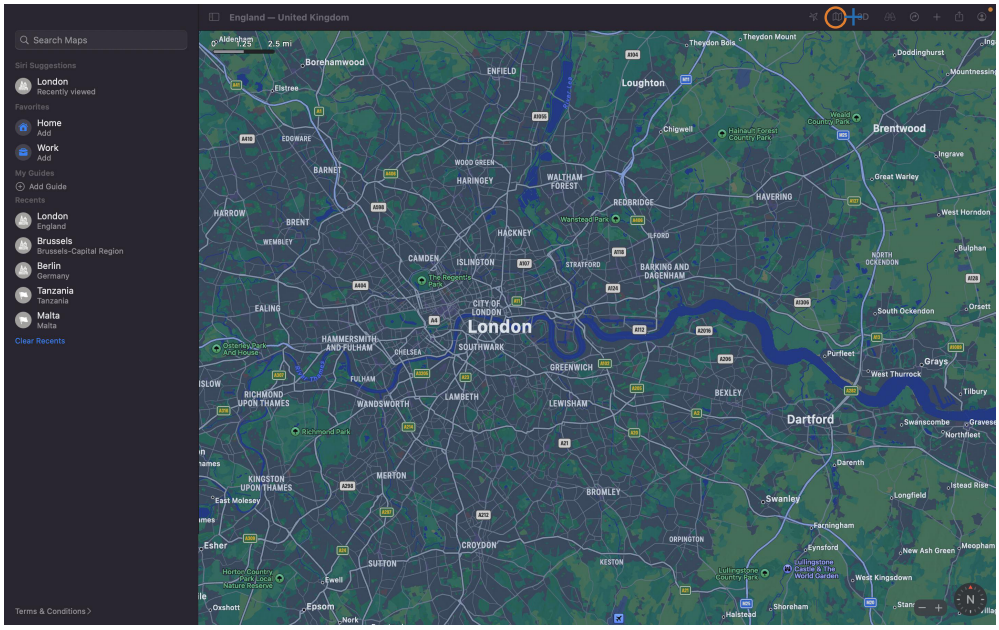
Source: OmniAct-Web, output index 0, dataset row 0.
Task: examine the items that have been added to the shopping trolley so far.
History: None.
Parsed SCA output: action = click; point = $(1276, 114)$.
Ground truth: action = click; point = $(1287, 115)$.
Evaluation: Correct action and accepted coordinate.



Orange ellipse: ground-truth target. Blue cross: parsed SCA coordinate.

Case 2: Desktop Map

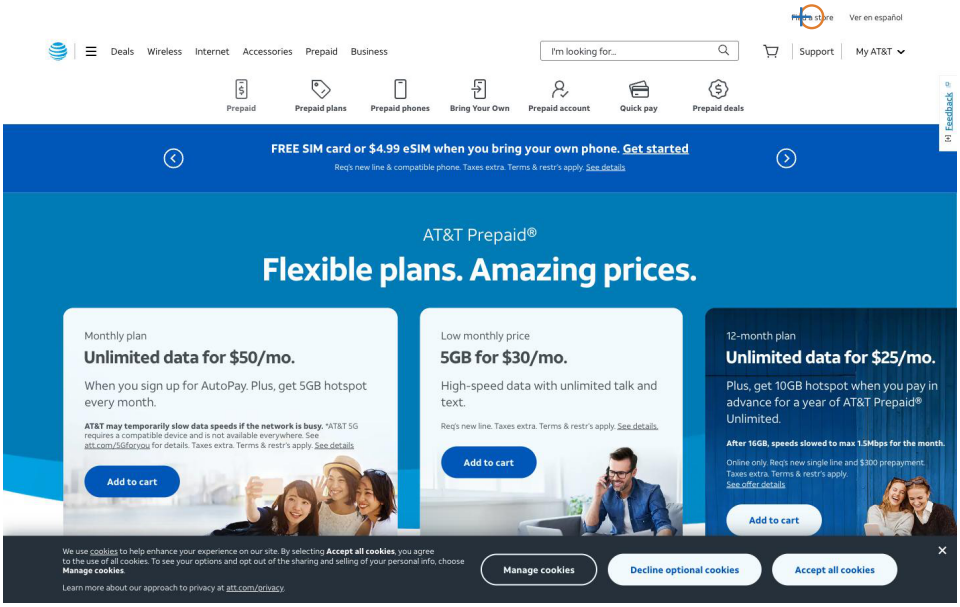
Source: OmniAct-Desktop, output index 8, dataset row 16.
Task: change the current map display to satellite view.
History: None.
Parsed SCA output: action = click; point = $(2465, 37)$.
Ground truth: action = click; point = $(2413, 39)$.
Evaluation: Correct action and accepted coordinate.



Orange ellipse: ground-truth target. Blue cross: parsed SCA coordinate.

Case 3: Web Double-Click Action

Source: OmniAct-Web, output index 546, dataset row 1092.
Task: execute double click function on `search_store_locations` tab.
History: None.
Parsed SCA output: action = doubleclick; point = $(1205, 14)$.
Ground truth: action = doubleclick; point = $(1221, 14)$.
Evaluation: Correct action and accepted coordinate.



Orange ellipse: ground-truth target. Blue cross: parsed SCA coordinate.

Case summary. The three cases cover a standard web click, a high-resolution desktop click, and a double-click action. They illustrate the prediction and evaluation interface; the rollout-level credit construction is shown separately by the deterministic mixed-group example in Appendix C.