GROUP-RELATIVE SELF-VERIFIED POLICY OPTIMIZATION FOR LABEL-FREE GUI GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for GUI grounding commonly relies on binary rewards that indicate whether a predicted click falls inside an annotated target region. Such rewards become uninformative when all rollouts in a group share the same outcome, and they also collapse near misses and distant errors into the same supervision. Our analysis further reveals that many near misses are produced with high confidence, suggesting that models often identify the approximate target region but lack sufficient clicking precision. These observations suggest that the missing supervision is not necessarily additional labels, but a graded signal that can exploit the model's own approximate localization. Motivated by this, we propose Group-Relative Self-Verified Policy Optimization (GSVPO), which converts local re-observation and cross-trajectory self-verification into a continuous, label-free grounding reward. GSVPO samples multiple global click proposals, re-examines each proposal under an entropy-adaptive magnified view, and maps the refined predictions back to the original coordinate space. Candidates are then evaluated according to their agreement with the ensemble of refined predictions across trajectories, preserving graded spatial distinctions without requiring ground-truth coordinates. The resulting signal provides a unified interface for training-free candidate reweighting, label-free test-time reinforcement learning, and reward augmentation for standard GRPO. Across six GUI grounding benchmarks and six vision-language models, training-free GSVPO improves 32 of 36 model–benchmark pairs, with an average gain of 6.03 percentage points.

1 INTRODUCTION

While CLI and code agents have advanced rapidly, graphical user interfaces remain indispensable for general-purpose computer use (Hu et al., 2026), making accurate visual grounding a core capability. A single misplaced click can invalidate an otherwise correct action plan, yet reliable grounding remains difficult in high-resolution interfaces containing tiny targets, and dense controls.

Reinforcement learning, particularly Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025), has recently become a common paradigm for training GUI grounding models (Luo et al., 2025; Lu et al., 2026; Sager et al., 2026; Yuan et al., 2025). Most existing methods rely primarily on sparse ground-truth rewards, typically a binary click reward indicating whether a predicted coordinate falls inside the annotated target region. Format or validity rewards may be added, but the core grounding signal remains the group-relative success–failure contrast. This creates an inherent degeneracy: when all rollouts hit the target or miss it, the binary grounding rewards become identical, and the grounding objective contributes no effective policy gradient.

Existing methods (Zhou et al., 2026; Luo et al., 2025; Liu et al., 2025a; Li et al., 2026a; Liu et al., 2025b) typically address this issue through *pre-training data filtering*. For example, only *mixed* groups, i.e., those with a success rate strictly between 0 and 1, are retained after sampling several clicks per example. However, GRPO is inherently on-policy: rollouts are repeatedly sampled from the current policy, whose distribution evolves throughout optimization. Consequently, an example that is mixed under the initial policy may soon become all-correct as the model improves, while stochastic decoding and policy shifts may also turn difficult examples into all-wrong groups. A one-time filter therefore cannot ensure informative gradients throughout training.

We empirically examine this limitation by simulating the pre-filtering procedure. From around 18000 candidate samples from UI-BERT (Bai et al., 2021), SeeClick (Cheng et al., 2024), and OS-Atlas (Wu et al., 2024), we use Qwen2.5-VL-3B (Bai et al., 2025) to sample eight clicks per sample

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.

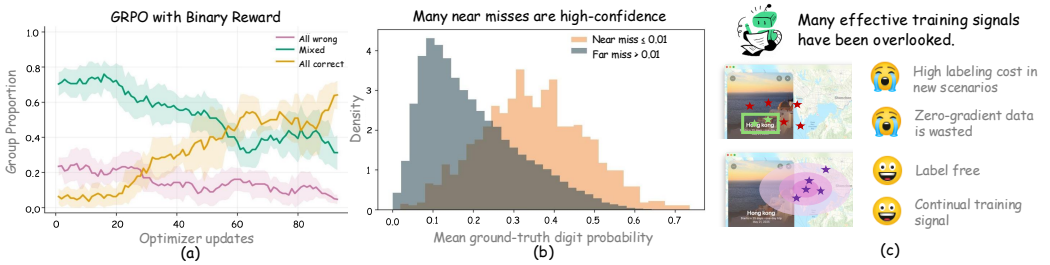


Figure 1: (a) shows the dynamic changes of pre-filtered training samples during training. (b) shows that incorrect clicks closer to the ground truth tend to have higher average confidence. A "near miss" is an incorrect click within 0.01 normalized screenshot coordinates of the nearest target-box boundary. (c) shows the advantages of the GSVPO reward design.

at temperature 1.0 and select 1,500 samples whose outcomes are initially mixed. We then train the model with binary-reward GRPO under the same sampling configuration and record the online group composition. As shown in Fig.1(a), the mixed-group ratio drops from 71% to 32%. Consequently, an increasing fraction of rollouts receives identical rewards and contributes no grounding gradient. This confirms that mixedness is a transient property of the current policy rather than an intrinsic property of an example, and that static filtering only postpones binary-reward saturation.

This naturally raises another question: when binary reward contrast vanishes, are such rollouts truly uninformative, or do they retain exploitable spatial structure? By analyzing the recorded 8-click test rollouts of candidate samples, we uncover a second, complementary finding. Incorrect clicks are not distributed uniformly: many lie close to the target boundary and are generated with substantially higher token-level confidence than distant errors (Fig. 1(b)). These predictions often identify the correct region but miss the target because of limited spatial precision, sometimes through only a small coordinate error. Binary reward treats such near misses exactly like unrelated clicks, discarding evidence that the model already knows approximately where the target is. This also highlights a limitation of relying solely on ground-truth label supervision: near target boundaries, small coordinate deviations can lead to sharply different rewards for visually similar clicks. These considerations motivate a complementary label-free training signal that provides continuous, within-group distinctions among candidate actions without requiring costly ground-truth bounding boxes. Therefore, since near misses may provide valuable evidence of approximate target localization, we ask whether revisiting each proposed region at higher visual resolution and self-verifying the resulting clicks can yield this training signal that without ground-truth supervision.

Humans rarely commit to a difficult click based solely on a single global observation; instead, they focus on a candidate region, inspect it at a finer scale, and verify the local evidence before acting. Motivated by these, we translate this verify-before-act behavior into a self-supervised group-wise mechanism. Specifically, we propose **GSVPO**, a **G**roup-Relative **S**elf-**V**erified **P**olicy **O**ptimization framework that constructs continuous grounding rewards without target-coordinate supervision. GSVPO first samples multiple proposals from the full screenshot. Each proposal defines a magnified local view, on which the same model grounds the instruction again; the refined coordinates are then mapped back to the original image. Crucially, GSVPO does not judge a proposal by whether it agrees only with its own refinement: an incorrect proposal may remain self-consistent after zooming and thereby reinforce its original error. Instead, every proposal is evaluated against the ensemble of locally verified coordinates from all trajectories. A candidate receives a high reward when it is supported by multiple independently initialized local observations, while isolated or trajectory-specific predictions receive lower rewards. This cross-trajectory agreement converts the model's sampling diversity into a source of supervision and yields a graded reward that preserves useful distinctions between near misses, plausible candidates, and clear outliers, even when binary hit-testing assigns them the same value. To make local verification reliable across targets of different sizes and uncertainty levels, GSVPO further uses the entropy of the coordinate-generation tokens to adapt the verification window: higher-uncertainty clicks are magnified more for a fine-grained view, while lower-uncertainty ones are magnified less to preserve visual context.

GSVPO's cross-trajectory self-verified advantages support three complementary uses: training-free logit reweighting with frozen parameters (Li et al., 2026b), label-free test-time reinforcement learning (Zuo et al., 2025), and reward augmentation for standard GRPO. Across six GUI grounding benchmarks and six vision-language models, GSVPO consistently improves grounding performance. One epoch of label-free adaptation on ScreenSpot-V2 (Wu et al., 2024) also transfers to

unseen benchmarks, while controlled training shows that the GSVPO reward alone outperforms binary supervision and that combining both rewards performs best.

Our contributions are threefold:

- We show that pre-filtered mixed examples do not remain informative during on-policy GRPO and identify confident near misses as revealing that the model may know the approximate target location but lack precise clicking capability, from which we attempt to create intrinsic, usable training signals.
- We introduce GSVPO, which converts entropy-guided cross-trajectory local verification into a continuous, label-free, group-relative grounding reward.
- We validate GSVPO as a training-free decoding method, a label-free test-time reinforcement learning objective, and a complementary reward for supervised GRPO across diverse models and benchmarks.

2 Related Work

Existing GUI grounding methods follow two complementary directions. Test-time enhancement methods (Lee et al., 2025; Hao et al., 2025; Nguyen, 2025) keep the backbone frozen and use additional computation to improve visual recognition and click precision. Training-based methods instead adapt a base model to GUI tasks through supervised fine-tuning (SFT) (Qin et al., 2025; Gou et al., 2025) or reinforcement learning (RL) (Chen et al., 2026; Zhao et al., 2026). Recent work has increasingly adopted reinforcement learning objectives, particularly GRPO.

Test-time enhancement. The prevailing strategy is to reduce visual redundancy by magnifying or cropping the region likely to contain the target (Pei et al., 2026). DiMo-GUI (Wu et al., 2025) progressively narrows attention through dynamic iterative zooming, whereas UI-Zoomer (Tang et al., 2026) treats both zoom triggering and crop scale as an uncertainty-estimation problem. GUI-RC (Du et al., 2025) aggregates multiple localization attempts through a spatial voting mechanism, and BAMI (Zhang et al., 2026) uses coarse-to-fine focusing and external candidate selection to improve click precision. These methods, however, either infer consistency primarily from full-screenshot predictions, which can preserve errors caused by tiny targets or visually similar distractors, or rely on a single zoom trajectory, making them vulnerable to trajectory-specific error amplification.

RL-based GUI grounding. Most GRPO-based methods like GUI-R1 (Luo et al., 2025), GUI-G1 (Zhou et al., 2026), InfiGUI-R1 (Liu et al., 2025a), and GUI-C^2 (Li et al., 2026a) use a binary reward; even with pre-filtering, gradient degradation (Fig.1(a)) still limits the performance ceiling as training scales. In contrast, GUI-RCPO (Du et al., 2025) and R-OPSD (Xuan & Li, 2026) move toward test-time reinforcement learning, obtaining continuous training signals without ground-truth coordinates. However, GUI-RCPO derives its reward from full-screenshot voting and therefore remains susceptible to shared distractor errors, whereas R-OPSD requires an additional reflector model in the optimization loop. Moreover, both methods leave the resolution bottleneck of extremely small targets on complex interfaces largely unaddressed (Liu et al., 2026).

GSVPO addresses the aforementioned limitations while remaining applicable to both test-time enhancement and test-time reinforcement learning.

3 Method

We consider a screenshot-instruction pair $x = (I, q)$ and a grounding policy $\pi_\theta(a \mid x)$ that emits a normalized click coordinate $a = (a_x, a_y) \in [0, 1]^2$. GSVPO replaces the external binary decision boundary with an *intrinsic verification process* derived from the policy's own spatial predictions. The central idea is to reinterpret additional model computation as a source of supervision rather than merely as an inference-time refinement mechanism. Concretely, it consists of three stages: (i) *grouped proposal generation*, which exposes multiple spatial modes of the current policy; (ii) *uncertainty-conditioned local verification*, which allocates visual resolution according to coordinate uncertainty; and (iii) *cross-trajectory consensus estimation*, which converts agreement among the resulting trajectories into a continuous group-relative preference.

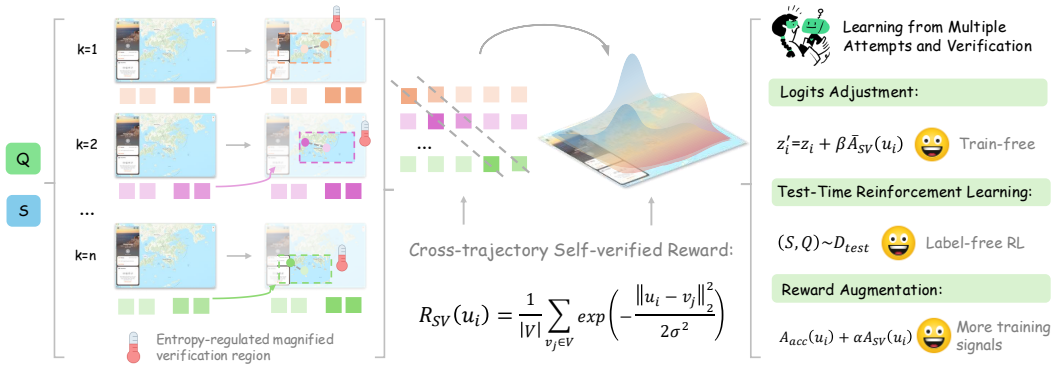


Figure 2: Overview of GSVPO. The model re-examines each proposed region in an entropy-guided magnified view. Cross-trajectory agreement provides a label-free reward for training-free reweighting, test-time reinforcement learning, and reward augmentation.

3.1 Grouped Spatial Hypothesis Generation

For each input x, we first construct a set of grounding hypotheses from the same policy that is subsequently optimized. We retain one deterministic mode of the policy through greedy decoding and sample the remaining hypotheses stochastically:

$$a_0 = \arg\max_a \pi_\theta(a \mid x), \qquad a_i \sim \pi_{\theta,T}(a \mid x), \quad i = 1, \ldots, K, \tag{1}$$

where T is the decoding temperature. We denote the resulting set by $\mathcal{A} = \{a_i\}_{i=0}^K$. Grouping is important for two reasons. It matches the relative-comparison structure of GRPO, and it attempts to reduce the risk that a single persistent mistake is promoted solely because it is internally consistent, by encouraging stochastic proposals.

3.2 Entropy-Conditioned Local Re-Observation

Proposal-conditioned re-observation. For each proposal a_i, we crop the screenshot around that coordinate, resize the crop to the model's visual-token budget, and ask the same policy to ground the instruction again:

$$I_i^{\text{loc}} = \mathcal{C}(I; a_i, w_i, h_i), \qquad \widetilde{v}_i = \arg\max_v \pi_\theta(v \mid I_i^{\text{loc}}, q). \tag{2}$$

The inverse crop transform M_i^{-1} maps the prediction back to the original coordinate system, $v_i = M_i^{-1}(\widetilde{v}_i)$. The verification set is $\mathcal{V} = \{v_i\}_{i=0}^K$. Boundary-aware padding is applied whenever a crop intersects the screenshot boundary, ensuring that the inverse transformation remains well-defined.

Uncertainty as a visual-resolution controller. A fixed crop is undesirable: a small window provides precision but can exclude the target, whereas a large window preserves context but weakens magnification. We therefore infer the window from the uncertainty of the coordinate tokens. Let $\mathcal{T}_x$ and $\mathcal{T}_y$ be the token positions used to decode the two coordinates, and let $p_t(d)$ be the policy distribution at position t. We accumulate axis-wise entropies

$$H_x = -\sum_{t \in \mathcal{T}_x} \sum_d p_t(d) \log p_t(d), \quad H_y = -\sum_{t \in \mathcal{T}_y} \sum_d p_t(d) \log p_t(d), \quad H = H_x + H_y. \tag{3}$$

The crop area decreases with uncertainty, within safe bounds:

$$S(H) = \text{clip}(S_0 - \gamma(H - H_0), S_{\min}, S_{\max}). \tag{4}$$

Thus an uncertain global click receives stronger magnification, while a confident click retains more context. Total entropy alone, however, does not capture directional ambiguity. We therefore additionally exploit the anisotropy between horizontal and vertical uncertainty:

$$\begin{aligned} \rho_H &= \left(\frac{H_x + \epsilon}{H_y + \epsilon}\right)^\eta, \\ w_i = \text{clip}\left(\sqrt{S(H)\rho_H}, w_{\min}, w_{\max}\right), &\quad h_i = \text{clip}\left(\sqrt{S(H)/\rho_H}, h_{\min}, h_{\max}\right). \end{aligned} \tag{5}$$

The window is elongated along the axis on which the model is less certain, while its total extent remains controlled. Consequently, uncertainty determines not only *how much* the model should zoom, but also *in which direction* additional visual evidence is most valuable. This couples the precision-context trade-off directly to an observable property of the current policy rather than to a fixed dataset-level heuristic.

3.3 Cross-Trajectory Self-Verification as Spatial Consensus

The pair (a_i, v_i) is a verification trajectory. Scoring a_i only by its paired v_i is insufficient: a proposal that is stably wrong after cropping would receive a high self-consistency score. GSVPO instead pools evidence from all local observations. Let $\mathcal{U} = \mathcal{A} \cup \mathcal{V}$ and define the Gaussian agreement kernel $k_\sigma(u, v) = \exp(-\|u - v\|_2^2/(2\sigma^2))$. The reward of any candidate $u \in \mathcal{U}$ is the empirical verification density

$$R_{\mathrm{sv}}(u) = \frac{1}{|\mathcal{V}|} \sum_{v_j \in \mathcal{V}} k_\sigma(u, v_j), \qquad \sigma = \kappa \max_i \{w_i, h_i\}. \tag{6}$$

The scale follows the verification resolution, making the score comparable across targets of different sizes. A candidate supported by several independently sampled local observations obtains a large density; an isolated prediction, including a trajectory-specific self-consistent error, receives a small density. This is a kernel-density view of consensus rather than a hard majority vote, so near misses remain ordered by their distance to the shared attractor.

We center and scale the reward within the complete group:

$$A_{\mathrm{sv}}(u) = \frac{R_{\mathrm{sv}}(u) - \mu_{\mathcal{U}}}{\sigma_{\mathcal{U}} + \epsilon}, \qquad \mu_{\mathcal{U}} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} R_{\mathrm{sv}}(u), \tag{7}$$

where $\sigma_{\mathcal{U}}$ is the corresponding standard deviation. The resulting A_{sv} transforms an absolute spatial-consensus score into a within-group preference in the same spirit as GRPO. Unlike a binary hit signal, this quantity can preserve relative distinctions among candidates even when all rollouts receive the same binary outcome.

3.4 A Unified Interface for Policy Integration

The construction above produces a single group-relative quantity, A_{sv}, that can be consumed either to select among candidates at inference time or to optimize the policy parameters. It is computed for every rollout group and is not gated on whether the binary reward is degenerate. For inference-time reweighting, if $z_i = \log \pi_\theta(u_i \mid x)$ is the sequence score, we form

$$z_i' = z_i + \beta A_{\mathrm{sv}}(u_i), \qquad q_{\mathrm{sv}}(u_i \mid x) = \frac{\pi_\theta(u_i \mid x) \exp(\beta A_{\mathrm{sv}}(u_i))}{\sum_j \pi_\theta(u_j \mid x) \exp(\beta A_{\mathrm{sv}}(u_j))}. \tag{8}$$

Selecting the candidate with the highest adjusted score applies the verified spatial preference without updating model parameters.

When parameter updates are permitted, the verified reward is used as the rollout return. In the label-free setting, a weighted policy objective (with the reward detached from back-propagation) is

$$\mathcal{L}_{\mathrm{sv}}(\theta) = -\frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} A_{\mathrm{sv}}(u_i) \log \pi_\theta(u_i \mid x). \tag{9}$$

An optional reference constraint, used when preserving the starting model is important, gives $\mathcal{L} = \mathcal{L}_{\mathrm{sv}} + \lambda D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$.

When target annotations are available, the same signal can be added as an auxiliary reward. For every rollout group, let A_{acc} denote the usual advantage of the binary hit reward and define

$$A_{\mathrm{tot}}(u_i) = A_{\mathrm{acc}}(u_i) + \alpha A_{\mathrm{sv}}(u_i). \tag{10}$$

This advantage is inserted into the same GRPO clipped surrogate used by the baseline, e.g.,

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{|\mathcal{U}|} \sum_i \min(r_i A_{\mathrm{tot}}(u_i), \mathrm{clip}(r_i, 1-\varepsilon, 1+\varepsilon) A_{\mathrm{tot}}(u_i)), \quad r_i = \frac{\pi_\theta(u_i \mid x)}{\pi_{\theta_{\mathrm{old}}}(u_i \mid x)}. \tag{11}$$

The experiments instantiate these interfaces as the three research questions in Section 4; the method itself is the shared self-verification signal rather than three separate algorithms.

3.5 A Density-Estimation View of Cross-Trajectory Verification

The self-verification mechanism admits a simple interpretation as latent spatial density estimation. If each local verification is correct with probability ρ and otherwise follows a more dispersed distribution, then the population version of Eq. equation 6 is a concentrated kernel around the latent target plus a lower-amplitude outlier term. Under a uniform (or sufficiently smooth) outlier component, its maximizer lies at (or in a neighborhood of) the target. With $K+1$ conditionally independent verification trajectories, Hoeffding's inequality (Hoeffding, 1963) gives an exponentially decreasing probability of reversing the ordering of two candidates whose population rewards differ by Δ: $\Pr[\widehat{R}(u_+) \le \widehat{R}(u_-)] \le 2\exp(-(K+1)\Delta^2/2)$. This guarantee is conditional rather than absolute, which also clarifies the limitation of label-free consensus: if all global proposals and all local re-observations collapse onto the same incorrect distractor, consensus can reinforce that shared error. GSVPO is therefore designed to mitigate single-trajectory error propagation and retain useful signal for near misses, not to recover every target without labels. Full proofs are discussed in Appendix A.1.

4 Experiment

We organize our experiments around three research questions that correspond to three complementary uses of GSVPO:

- **RQ1: Training-free inference.** Can GSVPO improve a frozen grounding policy by directly reweighting its decoding logits, without any parameter update?
- **RQ2: Test-time reinforcement learning.** Can the label-free reward support direct adaptation on an unlabeled test distribution, and does the resulting policy transfer beyond the adaptation benchmark?
- **RQ3: Reward design for policy optimization.** Can the self-verified reward train a grounding policy without ground-truth supervision, and is it complementary to the conventional binary reward when annotations are available?

4.1 Experimental Setup

Models, Benchmarks and Evaluation Metric. We evaluate performance on six benchmarks covering heterogeneous interfaces and difficulty levels: ScreenSpot-Pro (Li et al., 2025), ScreenSpot-V2, ScreenSpot (Cheng et al., 2024), MMBench-GUI (Wang et al., 2025), OSWorld-G, and OSWorld-G-R (Xie et al., 2025). For training-free evaluation, we use six GUI-capable vision-language models spanning different model families and scales: Qwen2.5-VL-3B, Qwen2.5-VL-7B (Bai et al., 2025), Qwen3.5-4B (Team, 2026), UGround-V1-7B (Gou et al., 2025), UI-TARS-1.5-7B (Qin et al., 2025), and UI-Venus-1.5-8B (Team et al., 2026). For test-time reinforcement learning and the controlled auxiliary reward comparison, we use Qwen2.5-VL family as the base policy. We follow the standard point-based grounding protocol and report grounding accuracy (%).

Implementation details. Unless otherwise specified, GSVPO samples $K = 8$ stochastic full-image proposals with temperature $T = 1.0$. Local verification is performed using the same model, and the resulting group-relative self-verification advantages are applied as described in Section 3. In RQ1, model parameters remain frozen and we add βA_{sv} directly to the proposal logits before selecting the final coordinate. In RQ2, we adapt the policy for one epoch on ScreenSpot-V2 using only the GSVPO reward and no ground-truth coordinates. In RQ3, all three variants use the same 1,500 examples, initialization, rollout budget, optimizer, and decoding configuration. Additional implementation details, including logit scale β, are provided in Appendix A.2.

4.2 RQ1: Training-Free Grounding via Logit Reweighting

To answer RQ1, we use the group-relative self-verification advantages to reweight the proposal logits and return the highest-scoring proposal. Table 1 shows improvements in 32 of 36 model-benchmark pairs, with a mean absolute gain of 6.03%. The improvement is especially pronounced on the more challenging benchmarks. On ScreenSpot-Pro, GSVPO improves Qwen2.5-VL-3B and Qwen2.5-VL-7B by 17.8% and 23.3%, respectively; on OSWorld-G-R, the corresponding gains are 13.3%

	SS-Pro	SS-V2	SS	MMB-GUI	OSW-G	OSW-G-R
Qwen2.5-VL-3B (Bai et al., 2025)	15.4	62.7	55.0	53.7	29.6	40.3
+ GSVPO	33.2	64.4	56.4	63.8	39.7	53.6
△ Ours	+17.8	+1.7	+1.4	+10.1	+10.1	+13.3
△ GUI-RC (Du et al., 2025)	+3.4	+2.5	+1.5	-	-	-
Qwen2.5-VL-7B (Bai et al., 2025)	26.9	88.2	84.6	60.9	33.9	42.9
+GSVPO	50.2	89.5	84.9	67.5	45.9	58.9
△ Ours	+23.3	+1.3	+0.3	+6.6	+12.0	+16.0
△ GUI-RC (Du et al., 2025)	+4.2	+2.0	+1.3	-	-	-
△ UI-Zoomer (Tang et al., 2026)	+13.4	-	-	-	-	-
Qwen3.5-4B (Team, 2026)	59.5	93.7	90.3	82.2	56.2	67.4
+GSVPO	72.4	93.0	90.4	85.8	64.0	75.9
△ Ours	+12.9	-0.7	+0.1	+3.6	+7.8	+8.5
UGround-V1-7B (Gou et al., 2025)	31.7	89.5	86.1	69.2	36.4	47.2
+GSVPO	44.3	90.1	86.8	72.5	46.1	58.3
△ Ours	+12.6	+0.6	+0.7	+3.3	+9.7	+11.1
△ GUI-RC (Du et al., 2025)	+0.1	+0.0	+0.2	-	-	-
UI-TARS-1.5-7B (Qin et al., 2025)	43.9	90.5	88.5	74.0	55.6	64.2
+GSVPO	56.6	90.1	88.3	74.7	60.1	68.8
△ Ours	+12.7	-0.4	-0.2	+0.7	+4.5	+4.6
△ GUI-RC (Du et al., 2025)	+0.3	+1.0	+0.8	-	-	-
△ BAMI (Zhang et al., 2026)	+11.1	+0.1	-	-	-	-
UI-Venus-1.5-8B (Team et al., 2026)	66.7	95.5	93.5	87.1	64.9	73.2
+GSVPO	74.0	95.7	93.4	87.5	65.4	76.1
△ Ours	+7.3	+0.2	-0.1	+0.4	+0.5	+2.9

Table 1: Performance improvements of different models across six benchmarks. SS-Pro, SS-V2, SS, MMB-GUI, OSW-G, and OSW-G-R denote ScreenSpot-Pro, ScreenSpot-V2, ScreenSpot, MMBench-GUI, OSWorld-G and OSWorld-G-Refined, and △ denotes the improvement of each method relative to the baseline.

and 16.0%. The method also transfers across architectures: it improves UGround-V1-7B by up to 12.6%, UI-TARS-1.5-7B by up to 12.7%, and UI-Venus-1.5-8B by 7.3% on ScreenSpot-Pro.

Gains are smaller on ScreenSpot and ScreenSpot-V2, where baseline accuracy reaches 95.5%. Across these two benchmarks, GSVPO produces four small regressions of at most 0.7%. This pattern suggests a ceiling effect: additional inference-time exploration is most useful when the original full-image policy is uncertain, while highly saturated benchmarks leave less room for correction. Overall, the consistent gains across model families and the absence of parameter updates support our central claim: cross-trajectory local verification rescues cases in which the model identifies the approximate target region but fails to click precisely, converting this self-knowledge into effective guidance signals. Consequently, the improvements are most pronounced for weak-precision-click models and more challenging benchmarks.

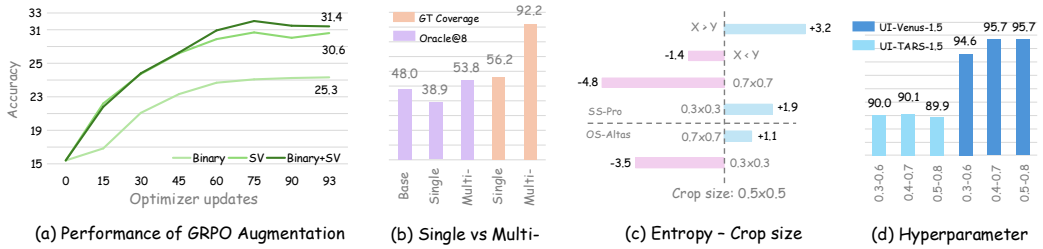
4.3 RQ2: Label-Free Test-Time Reinforcement Learning

At test time, GSVPO jointly updates all candidate click actions, including full-screenshot clicks and clicks conditioned on zoomed-in screenshots. We therefore evaluate both zoom-in and direct full-screenshot clicking. As shown in Table 2 (△ Ours represent zoom-in click), one epoch of GSVPO-TTRL improves both Qwen2.5-VL-3B and Qwen2.5-VL-7B on every benchmark. The 3B model gains 8.4% on the adaptation benchmark and an average of 12.7% across all six benchmarks, with particularly large gains on ScreenSpot-Pro (+13.4%), OSWorld-G (+19.5%), and OSWorld-G-R (+19.8%), despite receiving no updates from these datasets. The 7B model shows the same trend, achieving an average gain of 10.0% and improvements of 16.6%, 18.4%, and 17.4% on the same three benchmarks.

The direct-click results (w/ Direct Click) further show that these gains are not attributable solely to test-time cropping or magnification: self-verification on enlarged regions provides effective trajectory-level training signals for clicking directly on the original screenshot–something prior work has overlooked. Thus, GSVPO-TTRL does not merely memorize the ScreenSpot-V2 distribution; it learns a transferable preference for precise and visually consistent target locations, yielding improvements on unseen benchmarks. Compared with GUI-RCPO and R-OPSD, GSVPO-TTRL is competitive on ScreenSpot-V2 and provides larger gains on the more challenging OSWorld benchmarks, supporting the effectiveness of label-free test-time adaptation and transfer.

	SS-Pro	SS-V2	SS	MMB-GUI	OSW-G	OSW-G-R
Qwen2.5-VL-3B (Bai et al., 2025)	15.4	62.7	55.0	53.7	29.6	40.3
+ GSVPO-TTRL	28.8	71.1	62.4	61.3	49.1	60.1
△ Ours	+13.4	+8.4	+7.4	+7.6	+19.5	+19.8
△ GUI-RCPO (Du et al., 2025)	+4.5	+5.0	+4.0	+2.7	+2.2	+0.7
△ R-OPSD (Xuan & Li, 2026)	+10.2	+8.4	+7.6	+9.1	+5.5	+2.0
w/ Direct Click	+4.8	+8.9	+6.9	+1.0	+14.2	+12.9
Qwen2.5-VL-7B	26.9	88.2	84.6	60.9	33.9	42.9
+GSVPO-TTRL	43.5	89.4	86.1	66.0	52.3	60.3
△ Ours	+16.6	+1.2	+1.5	+5.1	+18.4	+17.4
△ GUI-RCPO (Du et al., 2025)	+6.1	+2.5	+2.4	-	-	-
w/ Direct Click	+1.6	+0.6	+1.3	+0.8	+13.8	+13.1

Table 2: Results of label-free test-time reinforcement learning on ScreenSpot-V2, with evaluation across six GUI grounding benchmarks.



(a) Performance of GRPO Augmentation (b) Single vs Multi- (c) Entropy - Crop size (d) Hyperparameter

Figure 3: (a) shows ScreenSpot-Pro accuracy of model checkpoints saved every 15 training steps. (b) shows the advantage of multi-view over single-view. (c) illustrates, through accuracy variation, how entropy magnitude and x-y axis differences regulate the crop box. (d) shows the effect of different total crop area ranges on the performance of strong grounding models.

4.4 RQ3: GSVPO as a Standalone and Complementary Training Reward

Fig. 3(a) shows that the label-free GSVPO reward alone consistently outperforms the conventional binary reward, reaching 30.6% versus 25.3% after 1,500 examples. The binary-only run improves initially but then saturates as an increasing fraction of groups become all-correct or all-wrong, eliminating within-group reward contrast. In contrast, GSVPO continues to assign graded advantages based on cross-trajectory spatial agreement and therefore preserves a useful learning signal even when binary hit outcomes are identical.

Combining the two rewards produces the strongest performance, reaching 31.4% and improving over GSVPO-only by 0.8%. The two signals are therefore complementary rather than redundant: the binary reward supplies precise task supervision whenever a group contains both successful and unsuccessful clicks, whereas GSVPO provides dense relative structure among predictions and remains active for binary-degenerate groups. As shown in Fig.3(a), when some samples begin to yield zero policy gradient under the binary reward in later training stages, the self-verified reward can further improve overall accuracy.

4.5 Ablation Study and Hyperparameter Study

Effectiveness of cross-trajectory multi-view. We conducted an eight-click rollout comparison on 1,000 OS-Atlas samples. In the first condition, we centered a $50\% \times 50\%$ crop on the first click and constrained all subsequent eight clicks to this fixed region. In the second condition, we constructed a separate crop around each of the eight clicks and generated one additional click within each crop; these eight post-zoom clicks were used as the final predictions. For both conditions, we measured ground-truth coverage and oracle@8. As shown in Fig. 3(b), cross-trajectory multi-view cropping achieves 92.2% ground-truth coverage and 53.8% oracle@8, whereas the single-view variant reaches only 56.2% coverage and 38.9% oracle@8, even below the 48.0% obtained by eight direct clicks without cropping. To further isolate the role of cross-trajectory verification, we computed the advantage using only the distance between the first click and its post-zoom click within each individual trajectory. As shown in Table 3, selecting the full-screen click with the highest single-trajectory consensus yields only 19.3% accuracy on ScreenSpot-Pro, while selecting the

corresponding zoom-in click achieves 26.1%. These results indicate that single-trajectory, single-view cropping at high temperature can readily exclude the ground-truth target from the crop region, leading to irrecoverable error propagation. Cross-trajectory multi-view cropping, although more expensive at inference time, mitigates this failure mode by providing diverse spatial perturbations and increasing the number of correct candidate clicks, thereby reducing the persistent errors associated with uncropped multi-click inference on the full screenshot.

Effectiveness of entropy-regulated crop region. We randomly selected 1,000 samples each from OS-Atlas and ScreenSpot-Pro, and examined how accuracy of Qwen2.5-VL-3B relates to entropy under varying crop sizes and x-y axis ratios as shown in Fig.3(c). On ScreenSpot-Pro (high average entropy), 0.3×0.3 crops improve accuracy by 1.9% over 0.5×0.5, while 0.7×0.7 drops by 4.8% relative to 0.5×0.5; on OS-Atlas (low average entropy), the opposite trend holds. This matches their qualitative properties: ScreenSpot-Pro samples are harder with extremely small targets, so the benefits of greater magnification outweigh those of retaining more visual context. Moreover, on ScreenSpot-Pro, keeping the total area constant while reshaping according to x-y entropy differences and extending the crop along the axis in which the model is more uncertain improves accuracy by 3.2%, while compressing along that axis drops it by 1.4%.

	SS-Pro
Qwen2.5-VL-3B (Baseline)	15.4
8 full-screenshot clicks in total	14.7
4-traj in total	29.0
Full-screenshot clicks as candidates	18.2
Zoom-in clicks as candidates	33.1
Single-traj full-screen click	19.3
Single-traj zoom-in click	26.1
GSVPO	33.2

Table 3: Performance under different candidate points and different self-verification trajectory settings.

Performance improvement comes more from effective self-verification. Under the training-free setting, using only the first full-screenshot click as the candidate while letting the zoom-in click contribute solely to reward computation still slightly improves accuracy to 18.2% on ScreenSpot-Pro, whereas using only the second zoom-in click as the candidate raises accuracy to 33.1%, almost matching the original GSVPO setting with all points as candidates as shown in Table3. This shows that for extremely small target tasks like ScreenSpot-Pro, candidates after magnification-based self-verification are more accurate and can rescue samples that already know the approximate location, suggesting that limited spatial precision is one important source of error for the evaluated model. To determine whether these gains arise from self-verification rather than simply from generating more candidates at a higher inference cost, we tested performance with only 4 trajectories and 8 clicks in total (4 full-screenshot clicks and 4 self-verification clicks). Results show that accuracy on ScreenSpot-Pro reaches 29.0%, close to the full GSVPO result, whereas considering only 8 full-screenshot clicks drops accuracy to 14.7%—the self-consistency from full screenshots instead introduces potentially erroneous minority trajectories that mislead clicks that should have hit.

Hyperparameter analysis of the crop region. We test the effect of different area constraint ranges. As shown in Fig.3(d), an area ratio of 0.3–0.5 means the crop occupies 0.3–0.5 of the full-screenshot and varies with entropy. Strong grounding models such as UI-TARS-1.5-7B and UI-Venus-1.5-8B show minimal performance variation on ScreenSpot-V2, whereas in Fig.3(c), crop ratio has a more pronounced impact on the weak grounding model Qwen2.5. This may be because the effect of crop area stems more from providing evidence for uncertain clicks, specifically, the difference between the fine-grained clicking and the full-screenshot clicking; weak models are more susceptible to visual context fluctuations, and precision level plays a significant role in correcting erroneous clicks. For such models, balancing precision and context is crucial.

5 Conclusion

We presented GSVPO, a group-relative self-verified policy optimization framework for label-free GUI grounding. GSVPO uses cross-trajectory local verification to construct label-free rewards that can distinguish candidates even when binary grounding rewards are identical. Specifically, it turns multiple global proposals into additional local observations, adapts the verification window to coordinate uncertainty, and derives an auxiliary reward from the spatial mutual agreement among candidates. Across training-free inference, label-free test-time adaptation, and reward-augmented policy optimization, experiments show consistent gains across models and benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools for provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation. We have not used generative AI tools for the rest of the required disclosure tasks like data generation are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper. We have reviewed all AI-assisted work. For our two main uses of AI tools: code rewriting and paper drafting and polishing, we conducted detailed reviews respectively. One author carefully read the rewritten code to ensure it is valid and consistent with the design described in the paper, and four authors carefully reviewed the polished text to ensure there is no fabricated content, and verified the formulas and theoretical analysis in detail. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work complies with ethical guidelines. All data involved in our research are open-source public data that have undergone reasonable privacy and security review, and these data are publicly accessible. In addition, our research does not involve any illegal, discriminatory, violent, or otherwise harmful impacts, and contains no harmful views.

REPRODUCIBILITY STATEMENT

The implementation of GSVPO conforms to the design and steps described in the main text. In AppendixA.2, we provide in detail all information needed for reproduction, including parameter designs, training designs, and decoding settings. In addition, we clearly specify all data sources and processing methods in the paper. Furthermore, for training-free GSVPO and GSVPO-TTRL, we provide clear algorithm pseudocode in the appendix.

REFERENCES

Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaise Aguera y Arcas. Uibert: Learning generic multimodal representations for ui understanding, 2021. URL https://arxiv.org/abs/2107.13731.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Chen Chen, Jiawei Shao, Dakuan Lu, Haoyi Hu, Xiangcheng Liu, Hantao Yao, and Wu Liu. Gui-eyes: Tool-augmented perception for visual grounding in gui agents, 2026. URL https://arxiv.org/abs/2601.09770.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents, 2024. URL https://arxiv.org/abs/2401.10935.

Yong Du, Yuchen Yan, Fei Tang, Zhengxi Lu, Chang Zong, Weiming Lu, Shengpei Jiang, and Yongliang Shen. Test-time reinforcement learning for gui grounding via region consistency, 2025. URL https://arxiv.org/abs/2508.05615.

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, 2025. URL https://arxiv.org/abs/2410.05243.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Chao Hao, Shuai Wang, and Kaiwen Zhou. Uncertainty-aware gui agent: Adaptive perception through component recommendation and human-in-the-loop refinement, 2025. URL https://arxiv.org/abs/2508.04025.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963. URL https://link.springer.com/chapter/10.1007/978-1-4612-0865-5_26.

Xuhao Hu, Xi Zhang, Haiyang Xu, Kyle Qiao, Jingyi Yang, Xuanjing Huang, Jing Shao, Ming Yan, and Jieping Ye. Toolcua: Towards optimal gui-tool path orchestration for computer use agents. *arXiv preprint arXiv:2605.12481*, 2026. URL https://arxiv.org/abs/2605.12481.

Hyunseok Lee, Jeonghoon Kim, Beomjun Kim, Jihoon Tack, Chansong Jo, Jaehong Lee, Cheonbok Park, Sookyo In, Jinwoo Shin, and Kang Min Yoo. Reguide: Data efficient gui grounding via spatial reasoning and search, 2025. URL https://arxiv.org/abs/2505.15259.

Junlong Li, Chao Hao, Lap-Pui Chau, and Yi Wang. Gui-c^2: Coarse-to-fine gui grounding via difficulty-aware reinforcement learning, 2026a. URL https://arxiv.org/abs/2605.30884.

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, 2025. URL https://arxiv.org/abs/2504.07981.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. Just-in-time reinforcement learning: Continual learning in llm agents without gradient updates, 2026b. URL https://arxiv.org/abs/2601.18510.

Yichao Liu, Huawen Shen, Liu Yu, Shiyu Liu, Zeyu Chen, and Yu Zhou. Drs-gui: Dynamic region search for training-free gui grounding, 2026. URL https://arxiv.org/abs/2605.15542.

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

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xavier Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization, 2025b. URL https://arxiv.org/abs/2508.05731.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Pengxiang Zhao, Guangyi Liu, et al. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17608–17616, 2026. URL https://arxiv.org/abs/2503.21620.

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. URL https://arxiv.org/abs/2504.10458.

Anthony Nguyen. Improved gui grounding via iterative narrowing, 2025. URL https://arxiv.org/abs/2411.13591.

Siqi Pei, Liang Tang, Tiaonan Duan, Long Chen, Shuxian Li, Kaer Huang, Yanzhe Jing, Yiqiang Yan, Bo Zhang, Chenghao Jiang, Borui Zhang, and Jiwen Lu. Adazoom-gui: Adaptive zoom-based gui grounding with instruction refinement, 2026. URL https://arxiv.org/abs/2603.17441.

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.

Pascal J. Sager, Benjamin Meyer, Peng Yan, Rebekka Von Wartburg-Kottler, Layan Etaiwi, Aref Enayati, Gabriel Nobel, Ahmed Abdulkadir, Benjamin F. Grewe, and Thilo Stadelmann. A comprehensive survey of agents for computer use: Foundations, challenges, and future directions. *Journal of Artificial Intelligence Research*, 85, March 2026. ISSN 1076-9757. doi: 10.1613/jair.1.19490. URL http://dx.doi.org/10.1613/jair.1.19490.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Fei Tang, Bofan Chen, Zhengxi Lu, Tongbo Chen, Songqin Nong, Tao Jiang, Wenhao Xu, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Ui-zoomer: Uncertainty-driven adaptive zoom-in for gui grounding, 2026. URL https://arxiv.org/abs/2604.14113.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Venus Team, Changlong Gao, Zhangxuan Gu, Yulin Liu, Xinyu Qiu, Shuheng Shen, Yue Wen, Tianyu Xia, Zhenyu Xu, Zhengwen Zeng, Beitong Zhou, Xingran Zhou, Weizhi Chen, Sunhao Dai, Jingya Dou, Yichen Gong, Yuan Guo, Zhenlin Guo, Feng Li, Qian Li, Jinzhen Lin, Yuqi Zhou, Linchao Zhu, Liang Chen, Zhenyu Guo, Changhua Meng, and Weiqiang Wang. Ui-venus-1.5 technical report, 2026. URL https://arxiv.org/abs/2602.09082.

Xuehui Wang, Zhenyu Wu, JingJing Xie, Zichen Ding, Bowen Yang, Zehao Li, Zhaoyang Liu, Qingyun Li, Xuan Dong, Zhe Chen, Weiyun Wang, Xiangyu Zhao, Jixuan Chen, Haodong Duan, Tianbao Xie, Chenyu Yang, Shiqian Su, Yue Yu, Yuan Huang, Yiqian Liu, Xiao Zhang, Yanting Zhang, Xiangyu Yue, Weijie Su, Xizhou Zhu, Wei Shen, Jifeng Dai, and Wenhai Wang. Mmbench-gui: Hierarchical multi-platform evaluation framework for gui agents, 2025. URL https://arxiv.org/abs/2507.19478.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning, 2025. URL https://arxiv.org/abs/2507.00008.

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, 2024. URL https://arxiv.org/abs/2410.23218.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis, 2025. URL https://arxiv.org/abs/2505.13227.

Shiyu Xuan and Zechao Li. Test-time self-evolving gui visual grounding via reflection-guided on-policy self-distillation, 2026. URL https://arxiv.org/abs/2608.11191.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning, 2025. URL https://arxiv.org/abs/2505.12370.

Borui Zhang, Bo Zhang, Bo Wang, Wenzhao Zheng, Yuhao Cheng, Liang Tang, Yiqiang Yan, Jie Zhou, and Jiwen Lu. Bami: Training-free bias mitigation in gui grounding, 2026. URL https://arxiv.org/abs/2605.06664.

Yu Zhao, Wei-Ning Chen, Huseyin Atahan Inan, Samuel Kessler, Lu Wang, Lukas Wutschitz, Fangkai Yang, Chaoyun Zhang, Pasquale Minervini, Saravan Rajmohan, and Robert Sim. Learning gui grounding with spatial reasoning from visual feedback, 2026. URL https://arxiv.org/abs/2509.21552.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. *Advances in Neural Information Processing Systems*, 38:95683–95705, 2026. URL https://arxiv.org/abs/2505.15810.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning, 2025. URL https://arxiv.org/abs/2504.16084.

A Appendix

A.1 Theoretical analysis of cross-trajectory self-verification

We formalize the intuition in Section 3.5. Fix an input x and let $a^\star \in [0,1]^2$ be its unobserved target location. A verification point is modeled as

$$V \sim \rho\mathcal{N}(a^\star, \Sigma_v) + (1-\rho)Q_{\text{out}}, \qquad 0 < \rho \le 1, \tag{12}$$

where successful local observations are concentrated around $a^\star$ and Q_{out} describes failed or distractor verifications. The assumption does not require every trajectory to succeed; it only requires successful observations to be more concentrated than failures. Let $k_\sigma(u,v) = \exp(-\|u-v\|_2^2/(2\sigma^2))$ and define the population score

$$m(u) = \mathbb{E}_V[k_\sigma(u,V)]. \tag{13}$$

Proposition 1 (latent-target recovery). Under Eq. equation 12,

$$m(u) = \rho c \exp\left[-\frac{1}{2}(u-a^\star)^\top(\Sigma_v+\sigma^2 I)^{-1}(u-a^\star)\right] + (1-\rho)b(u), \tag{14}$$

where

$$c = \left|I + \Sigma_v/\sigma^2\right|^{-1/2}, \qquad b(u) = \mathbb{E}_{V\sim Q_{\text{out}}}[k_\sigma(u,V)]. \tag{15}$$

If $b(u)$ is constant, $m(u)$ is uniquely maximized at $a^\star$. More generally, if the variation of $(1-\rho)b(u)$ is smaller than the decrease of the Gaussian term outside a neighborhood $\mathcal{N}(a^\star)$, every maximizer of m lies in that neighborhood.

Proof. The Gaussian component is the convolution of the verification Gaussian with the kernel Gaussian. The standard Gaussian convolution identity yields the first term of Eq. equation 14, whose unique maximum is $u = a^\star$. A uniform outlier component contributes a constant and hence does not change the maximizer. For a non-uniform component, the maximizer can move only by the amount of the outlier perturbation; the stated margin condition ensures that this perturbation cannot overcome the decrease of the concentrated component outside $\mathcal{N}(a^\star)$.

Proposition 2 (finite-group concentration). For $K+1$ conditionally independent verification points $V_0, \dots, V_K$, define

$$\widehat{R}_K(u) = \frac{1}{K+1}\sum_{j=0}^{K} k_\sigma(u, V_j). \tag{16}$$

Since $k_\sigma \in [0,1]$, Hoeffding's inequality gives, for any $\delta > 0$,

$$\Pr\left(\left|\widehat{R}_K(u) - m(u)\right| \ge \delta\right) \le 2\exp\left(-2(K+1)\delta^2\right). \tag{17}$$

For candidates u_+ and u_- with population gap $m(u_+) - m(u_-) = \Delta > 0$,

$$\Pr\left(\widehat{R}_K(u_+) \le \widehat{R}_K(u_-)\right) \le 2\exp\left(-\frac{(K+1)\Delta^2}{2}\right). \tag{18}$$

Proof. Apply Eq. equation 17 with $\delta = \Delta/2$ to each candidate and use a union bound. An empirical ordering reversal implies that at least one of the two estimates deviates from its population value by at least $\Delta/2$, giving Eq. equation 18 after a union bound.

Setting	Qwen2.5-VL-3B	Qwen2.5-VL-7B
Training samples	1,272	1,272
Epochs	1	1
Optimizer	AdamW	AdamW
Learning rate	5×10^{-7}	5×10^{-7}
Adam betas	$(0.9, 0.95)$	$(0.9, 0.95)$
Per-GPU batch size	4	2
Gradient accumulation	1	2
Effective batch size	16	16
Optimizer updates	79	80
Warm-up updates	8	8
LR schedule	cosine decay	cosine decay
Gradient clipping	1.0	1.0
Precision	BF16	BF16
Gradient checkpointing	enabled	enabled
Attention kernel	FlashAttention-2	FlashAttention-2
KL regularization coefficient	0	0
GPUs	4×A800-80GB	4×A800-80GB

Table 4: Training configuration for GSVPO-TTRL on ScreenSpot-V2.

Implication for GSVPO. A single-trajectory score $k_\sigma(a_i, v_i)$ can be large when a proposal reproduces its own error and therefore cannot separate a correct attractor from a trajectory-specific one. Cross-trajectory aggregation estimates the shared density $m(u)$ instead. By Proposition 1, the concentrated successful component pulls the population maximum toward the latent target; by Proposition 2, the probability that a finite group reverses the ordering of two candidates decreases exponentially in the number of verification trajectories. This is the formal basis for using A_{sv} as a continuous, label-free group-relative learning signal.

Scope and failure mode. The propositions require that a non-zero fraction of local verifications be more concentrated near the latent target than the outlier component. They do not guarantee recovery when all global proposals and their local refinements collapse onto a common distractor: in that case the empirical verification density itself has its largest mode at the distractor. This is an intrinsic identifiability limitation of label-free consensus, not a contradiction of the method. Cross-trajectory verification reduces the risk of isolated or trajectory-specific errors, while proposal diversity and entropy-adaptive windows determine whether at least some trajectories escape a shared failure.

A.2 Details in Experimental Setting

In train-free inference setting (**RQ1**), the coordinate generation uses at most 32 new tokens and an image budget of 99,999,999 pixels. The crop window is determined from the greedy prediction's coordinate-token entropy: its area is linearly adapted within $0.04, 0.25$ for ScreenSpot-Pro, OSWorld-G, and OSWorld-G-R (ScreenSpot-V2, ScreenSpot, and MMBench-GUI are set to $0.4, 0.7$ since they have bigger targets), while its aspect ratio follows the entropy ratio with exponent 0.25; crop side lengths are clipped to $0.10, 0.85$. The spatial-consensus kernel uses the relative bandwidth $\sigma_0 = k \max(w, h)$ with $k = 0.1$. Candidate sequence scores are reweighted by $z_i' = z_i + \beta A_{\mathrm{sv}}(u_i)$, with $\beta = 1.0$ by default, and the final prediction is selected from the complete candidate pool. Seed is set to 42.

In GSVPO-TTRL setting (**RQ2**), we perform label-free test-time reinforcement learning on the complete ScreenSpot-v2 collection (1,272 samples) for one epoch, without a validation split or access to ground-truth coordinates during optimization. Both runs use the same whole-trajectory GRPO implementation and differ only in backbone size and per-GPU micro-batch configuration. The principal settings are listed in Table 4.

For each training example, the policy samples eight full-image proposals at temperature 1.0 and performs one local re-click for each proposal, forming a 16-candidate group. A separate greedy prediction is used only to estimate coordinate-token entropy and determine the adaptive verification window; it is not included as a trainable candidate. Using the entropy of a greedy full-image predic-

tion to determine the crop geometry provides a stable, sample-level estimate of visual uncertainty. It separates uncertainty estimation from spatial exploration: the greedy prediction controls the crop size and aspect ratio, while stochastic proposals determine the crop centers. This shared geometry makes local verification across trajectories more comparable and avoids introducing additional variability or candidate-dependent computational advantages that could confound the self-verified reward. The reward scale is $\beta = 0.2$ and the spatial kernel coefficient is $k = 0.1$. We use nucleus sampling with $p = 0.9$, a maximum of 32 generated tokens, the test-time image budget of 99,999,999 pixels, and random seed 42. The models are trained with four-process distributed data parallelism and DeepSpeed ZeRO-2. The warm-up length is set to 10% of the total optimizer updates (eight updates in both runs), followed by cosine decay.

For **RQ3**, we train Qwen2.5-VL-3B for one epoch on 1,500 training examples using groups of eight stochastic rollouts per input. The accuracy and self-verification rewards are independently normalized within the complete base-plus-zoom candidate pool and combined as

$$A_{\mathrm{tot}}(u_i) = A_{\mathrm{acc}}(u_i) + \alpha A_{\mathrm{sv}}(u_i), \qquad \alpha = 1.$$

In the implementation, this is realized as the equivalent equal-weight mixture

$$0.5A_{\mathrm{acc}}(u_i) + 0.5A_{\mathrm{sv}}(u_i),$$

which differs only by a global scale factor. We use a learning rate of 5×10^{-7} with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$), cosine learning-rate decay, and approximately 10% warm-up steps. The policy-gradient loss is scaled by $\beta = 0.2$, with no KL regularization ($\beta_{\mathrm{KL}} = 0$). The per-device batch size is 4 with no gradient accumulation. Rollouts use temperature 1.0, top-$p = 0.9$, and at most 32 generated tokens. Self-verification uses a kernel scale $k = 0.1$. All experiments use random seed 42.

Training-dynamics analysis. To analyze the evolution of binary-reward groups in Fig. 1, we additionally train Qwen2.5-VL-3B for one epoch on the pre-filtered 1,500-example mixed set. This matched binary-GRPO run uses eight stochastic full-image rollouts per example, per-GPU batch size 4, learning rate 5×10^{-7}, AdamW with betas $(0.9, 0.95)$, cosine decay with a 10% warm-up, and seed 42. Unlike the GSVPO-TTRL run, it uses only the ground-truth binary hit/miss reward and no greedy anchor or local zoom trajectory; group composition and policy-gradient statistics are logged online throughout training.

For the approximately 18,000 samples used as data sources, we directly selected data from UI-BERT, SeeClick, and OS-Atlas, since this part of the experiment serves as an analytical experiment and we consider it sufficient to support validation of the sample dynamics during training.

Algorithm 1: Group-Relative Self-Verified Policy Optimization

Require: Screenshot-instruction pair $x = (I, q)$; policy π_θ; number of stochastic proposals K; temperature T; kernel coefficient κ; scale β

Ensure: A final candidate or a policy-update loss

1: Obtain a greedy prediction and its coordinate-token distributions.
2: Compute the axis-wise entropies (H_x, H_y) and determine the verification window (w, h).
3: Sample K full-screenshot proposals:
$\mathcal{A}_{\mathrm{s}} = \{a_i\}_{i=1}^{K}, \quad a_i \sim \pi_{\theta,T}(\cdot \mid x)$.
4: Construct the base candidate set $\mathcal{A}$. For training-free inference, the greedy prediction may also be included.
5: **for all** $a_i \in \mathcal{A}$ **do**
6: Crop and resize a local view of size (w, h) around a_i.
7: Re-localize the instruction in the local view and map the result back to the original coordinate system:
$v_i = M_i^{-1}(\mathrm{Decode}_{\pi_\theta}(C(I; a_i, w, h), q))$.
8: **end for**
9: Form the verification set $\mathcal{V} = \{v_i\}$ and the complete candidate pool $\mathcal{U} = \mathcal{A} \cup \mathcal{V}$.
10: Set the spatial bandwidth:
$\sigma = \kappa \max(w, h)$.
11: **for all** $u \in \mathcal{U}$ **do**

Compute the cross-trajectory verification density:

$$R_{\mathrm{sv}}(u) = \frac{1}{|\mathcal{V}|} \sum_{v_j \in \mathcal{V}} \exp\left(-\frac{\|u - v_j\|_2^2}{2\sigma^2}\right).$$

end for

Normalize the reward over the complete candidate pool:

$$A_{\mathrm{sv}}(u) = \frac{R_{\mathrm{sv}}(u) - \mu_{\mathcal{U}}}{\sigma_{\mathcal{U}} + \epsilon}.$$

if training-free inference **then**

Compute $z_i' = \log \pi_\theta(u_i \mid x) + \beta A_{\mathrm{sv}}(u_i)$.

Return $u^\star = \arg\max_{u_i \in \mathcal{U}} z_i'$.

else if label-free test-time reinforcement learning **then**

Detach A_{sv} from back-propagation.

Optimize $\mathcal{L}_{\mathrm{sv}} = -\frac{\beta}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} A_{\mathrm{sv}}(u_i) \log \pi_\theta(u_i \mid x)$.

end if

A.3 Limiting Behavior of the Spatial Bandwidth

The kernel coefficient κ controls the spatial tolerance of the self-verification reward through

$$\sigma = \kappa \max(w, h).$$

Thus, κ determines how strictly candidate locations must agree with the verified predictions. Its limiting behavior clarifies the trade-off between spatial precision and reward smoothness.

When $\kappa \to 0$, the Gaussian kernel becomes sharply concentrated:

$$k_\sigma(u, v) = \exp\left(-\frac{\|u - v\|_2^2}{2\sigma^2}\right) \longrightarrow \begin{cases} 1, & u = v, \\ 0, & u \neq v. \end{cases}$$

The resulting reward approaches a hard neighborhood-counting rule. This provides strong local discrimination, but candidates that are only slightly separated may receive nearly identical zero rewards, making the signal sensitive to coordinate noise and reducing its ability to rank near misses.

At the opposite extreme, when $\kappa \to \infty$,

$$k_\sigma(u, v) \approx 1 - \frac{\|u - v\|_2^2}{2\sigma^2},$$

and all candidates receive nearly identical rewards. Consequently, $\mathrm{std}_{u \in \mathcal{U}}[R_{\mathrm{sv}}(u)]$ approaches zero and the group-relative advantage loses discriminative power after normalization. Therefore, an excessively large bandwidth effectively removes the spatial preference learned from cross-trajectory agreement.

GSVPO operates in the intermediate regime, where the bandwidth is large enough to preserve a smooth ordering among nearby candidates but small enough to distinguish separate spatial modes. Scaling σ by the verification window size makes this tolerance relative to the visual resolution of the local observation. In all experiments, we use $\kappa = 0.1$ unless otherwise specified.

A.4 Group Normalization of the Self-Verification Reward

The raw spatial density $R_{\mathrm{sv}}(u)$ depends on the number and spatial spread of verification points and is therefore not directly comparable across different rollout groups. We convert it into a group-relative advantage by normalizing over the complete candidate pool $\mathcal{U} = \mathcal{A} \cup \mathcal{V}$:

$$A_{\mathrm{sv}}(u) = \frac{R_{\mathrm{sv}}(u) - \mu_{\mathcal{U}}}{\sigma_{\mathcal{U}} + \epsilon},$$

where

$$\mu_{\mathcal{U}} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} R_{\mathrm{sv}}(u), \qquad \sigma_{\mathcal{U}}^2 = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \big(R_{\mathrm{sv}}(u) - \mu_{\mathcal{U}}\big)^2.$$

The centering operation removes group-specific reward offsets, while positive rescaling does not change the ordering of candidates. Standardization also makes the magnitude of the self-verification signal comparable across inputs with different numbers of valid re-clicks, crop sizes, and spatial dispersions. The small constant $\epsilon = 10^{-6}$ prevents numerical instability when the group has nearly identical rewards.

Importantly, normalization is performed over the complete base-plus-zoom pool rather than separately over the two candidate types. Consequently, the advantage directly compares full-screenshot proposals with their locally verified counterparts. A candidate receives a positive advantage when its verification density is above the group mean and a negative advantage when it falls below the group mean. This preserves the group-relative interpretation required by the training-free logit update and by the policy-gradient objective.