SEEING THE FOREST AND THE TREES: MULTI-SCALE VOTING IMPROVES TEST-TIME GUI GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

GUI grounding, the localization of screen elements based on natural language instructions, is essential for autonomous GUI agents. Supervised training achieves strong accuracy but requires massive labeled data, making annotation expensive and limiting generalization to new environments. Recent work addresses this challenge by exploring test-time grounding, which exploits unlabeled data available during inference via iterative cropping and spatial voting. In this paper, we identify scale variation as an important yet overlooked factor in test-time GUI grounding and propose a simple Multiscale Spatial Voting (`MSV`) method. Specifically, `MSV` scales the full image up and down to multiple resolutions, generates multiple predictions at each scale, and aggregates these predictions via spatial voting to produce the final output. We integrate `MSV` into various test-time grounding approaches, including three training-free methods and two training-based methods. Extensive experiments on four datasets with various backbones confirm that `MSV` consistently improves grounding performance. Despite its simplicity, we believe this work provides a promising path toward label-efficient GUI agents.

1 INTRODUCTION

Graphical user interfaces (GUIs) are the primary medium through which users interact with digital devices, comprising elements such as buttons, menus, and text fields. As autonomous agents are increasingly deployed to perform tasks on behalf of users (Hong et al., 2024; Wang et al., 2024b), they must accurately map a natural language instruction to the corresponding screen element before executing any action. This capability, known as GUI grounding, is a foundational component of GUI agents, since errors at this stage propagate directly to downstream planning and execution.

Existing GUI grounding approaches fall into two paradigms according to their prediction type. The first casts grounding as point prediction, where the model directly outputs the target coordinate, as in SeeClick (Cheng et al., 2024) and UGround (Gou et al., 2024). The second predicts a bounding box that delineates the target region, as in OS-Atlas (Wu et al., 2025c) and GUI-Actor (Wu et al., 2025b). Point-based prediction integrates seamlessly with downstream actions but often fails to capture the spatial ambiguity inherent in human-GUI interaction, which has motivated region-based alternatives. However, both paradigms achieve strong performance primarily through extensive labeled data. Some methods rely on supervised fine-tuning, directly learning to predict coordinates or bounding boxes from annotated pairs (Qin et al., 2025; Liu et al., 2026). Others use reinforcement learning (RL) with reward signals computed from ground-truth labels (Liu et al., 2025; Luo et al., 2025a; Lu et al., 2026). Either way, labeled data is indispensable, and this reliance limits applicability in label-scarce scenarios, where large-scale GUI annotations are costly to collect.

To reduce this dependence, test-time scaling (Snell et al., 2024; Muennighoff et al., 2025) has recently gained popularity in GUI grounding (Luo et al., 2025b; Wu et al., 2025a; Zhang et al., 2026; Du et al., 2026), improving inference-time performance through additional computation rather than extra labels. For example, RegionFocus (Luo et al., 2025b) predicts per sub-region and aggregates into one refined action, while MVP (Zhang et al., 2026) crops sub-regions and returns the largest cluster centroid of predicted coordinates. In contrast to training-free methods, GUI-RCPO (Du et al., 2026) offers a training-based solution that enables test-time RL through region consistency signals as rewards for policy optimization. However, existing methods either crop sub-regions from the original screenshot or sample repeatedly at a single scale, which is insufficient for GUI screenshots

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.

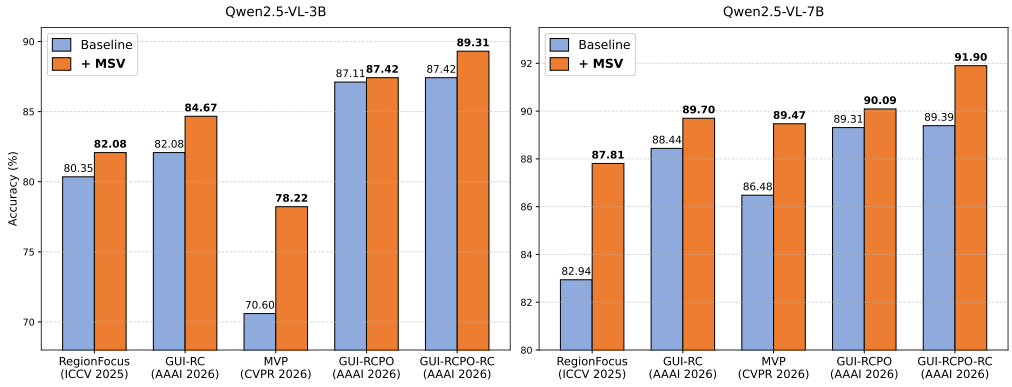

Figure 1: Results on the ScreenSpot-V2 dataset with Qwen2.5-VL-Instruct models. MSV consistently improves performance when integrated with existing GUI grounding methods.

where target elements vary drastically in size, from tiny icons on high-resolution professional interfaces to large buttons on mobile screens. At a single scale, the model's errors are often correlated rather than diverse: a small icon may be consistently mislocalized simply because it loses resolution once the full screenshot is encoded. Voting over such correlated errors then reinforces the mistake rather than correcting it.

To address this, we propose MSV, a simple yet effective multi-scale spatial voting method. It applies up- and down-scaling to the full screenshot to generate multiple views at different resolutions, then samples predictions from each view and aggregates them via spatial voting. Unlike prior cropping-based approaches, we resize the entire image. This design preserves the complete spatial context across all views, ensuring that the model perceives the full layout and avoids excluding the target element. To validate the effectiveness, we respectively integrate MSV with several popular test-time grounding methods, *i.e.*, RegionFocus (Luo et al., 2025b), MVP (Zhang et al., 2026) and GUI-RCPO (Du et al., 2026), and observe consistent performance gains, as shown in Figure 1.

To summarize, our contributions are as follows. First, we identify scale variation as an important yet overlooked factor in test-time GUI grounding, and propose MSV, a simple multi-scale spatial voting method that resizes the full screenshot to multiple resolutions and aggregates predictions via spatial voting. Second, extensive experiments on four datasets and various backbones show that MSV consistently improves training-free test-time grounding approaches, offering a promising path toward label-efficient GUI agents. Third, MSV also boosts performance when combined with test-time RL, and yields additional gains when applied on the model updated by test-time RL.

2 Related Work

2.1 GUI Grounding

GUI grounding is a foundational capability for GUI agents, enabling the translation of natural language instructions into executable interface actions such as clicking, typing, and navigation. Different from Qwen-VL models (Wang et al., 2024a) that perform GUI grounding as general-purpose Multimodal Large Language Models (MLLMs), SeeClick (Cheng et al., 2024) performs fine-tuning with labeled GUI examples that pair screenshots and instructions with target-element coordinates, using LoRA. Similarly, other methods (Lin et al., 2025; Xu et al., 2025; Wu et al., 2025c; Qian et al., 2026) have also relied on supervised fine-tuning to train models on labeled UI datasets. For example, ShowUI (Lin et al., 2025) introduces UI-aware visual token selection to better align vision, language, and action. UGround (Qian et al., 2026) leverages large-scale synthetic data to boost visual grounding, while OS-Atlas (Wu et al., 2025c) builds a large cross-platform grounding corpus.

More recently, inspired by the DeepSeek-R1 paradigm (Guo et al., 2025), RL-based fine-tuning has been introduced to GUI grounding. UI-R1 (Lu et al., 2026) and GUI-R1 (Luo et al., 2025a) adopt binary rule-based rewards for policy optimization. SE-GUI (Yuan et al., 2025) introduces a

self-evolutionary RL framework with dense policy gradients, while GUI-G^2 (Tang et al., 2026) replaces binary rewards with continuous Gaussian distributions for richer spatial supervision. InfiGUI-G1 (Liu et al., 2026) improves RL exploration through adaptive exploration policy optimization, and GuirlVG (Kang et al., 2026) stabilizes RL fine-tuning via an adversarial KL factor. Beyond reward design, COPC (Wang et al., 2026) shifts GUI grounding from a single-pass regression task to a learnable visual perception ranking paradigm and proposes a co-evolutionary RL strategy, while UI-Ins (Chen et al., 2026) reformulates instructions as dynamic reasoning pathways and adopts a two-stage framework in which SFT instills multi-perspective reasoning and RL optimizes pathway selection and composition. Despite these advances, most existing methods overlook a fundamental mismatch: the generative MLLM paradigm is ill-suited to the fine-grained spatial alignment demanded by point-level GUI grounding.

2.2 Test-time Scaling

Test-time scaling dynamically adjusts computational resources during inference to enhance model performance (Wang et al., 2023; Snell et al., 2024; Muennighoff et al., 2025; Zhang et al., 2025; Zuo et al., 2025; Yu et al., 2026). Inspired by its success in LLMs, several studies have extended similar principles to GUI grounding. On one hand, Nguyen (2024); Luo et al. (2025b); Wu et al. (2025a); Liu et al. (2026) explore training-free paradigms that leverage test-time scaling (Sample-K) or self-reflection for refinement. For example, Iterative-Narrowing (Nguyen, 2024) repeatedly crops around an MLLM's predicted GUI element to improve grounding precision. RegionFocus (Luo et al., 2025b) proposes focal points to extract targeted sub-regions, predicts actions individually for these sub-regions, and aggregates them into a single refined action for standard inference. DiMo-GUI (Wu et al., 2025a) treats text and icon modalities separately and progressively refines the focus region via dynamic zooming, after which a decision maker selects the final candidate from the text-based or icon-based outputs. MVP (Zhang et al., 2026) generates multiple views by cropping sub-regions from the original screenshot, and performs spatial clustering on all predicted coordinates to return the largest cluster's centroid. On the other hand, several other methods allow model update at test time. GUI-RCPO (Du et al., 2026) constructs spatial voting grids from multiple sampled predictions to identify consensus regions where models show the highest agreement. R-OPSD (Xuan & Li, 2026) proposes a reflection-guided test-time self-evolving framework that uses an MLLM-based evaluator and on-policy self-distillation with contrastive calibration to convert textual feedback from failed GUI grounding attempts into token-level supervision for post-deployment adaptation. To the best of our knowledge, this is the first work to investigate the importance of scaling over the original screenshot. Moreover, the proposed multi-scale voting technique is agnostic to both training-free and training-based grounding methods.

3 Method

We first describe the test-time GUI grounding setup. We then present multi-scale spatial voting (`MSV`), our simple yet effective approach. Subsequently, we integrate `MSV` with three training-free methods and one training-based method, and explain the details of each combination.

3.1 Problem Formulation

GUI grounding is the task of determining where a natural language instruction refers to on a graphical interface. Let $I \in \mathbb{R}^{H\times W\times 3}$ denote a screenshot and ins an instruction, and let M be a model that predicts the location of the UI element best described by ins. Prior work has explored two main output formats for this task:

$$\begin{aligned} \text{Point-based:} \quad & M(I, \text{ins}) \to (x, y), \\ \text{Region-based:} \quad & M(I, \text{ins}) \to (x_1, y_1, x_2, y_2). \end{aligned} \tag{1}$$

Unless otherwise specified, we use the region-based format throughout this work.

A representative training-free method, RegionFocus (Luo et al., 2025b), proposes focal points to extract targeted sub-regions, predicts actions individually for these sub-regions, and aggregates them into a single refined action for standard inference. By contrast, training-based methods such as GUI-RCPO (Du et al., 2026) are inspired by TTRL (Zuo et al., 2025; Yan et al., 2026), a paradigm that

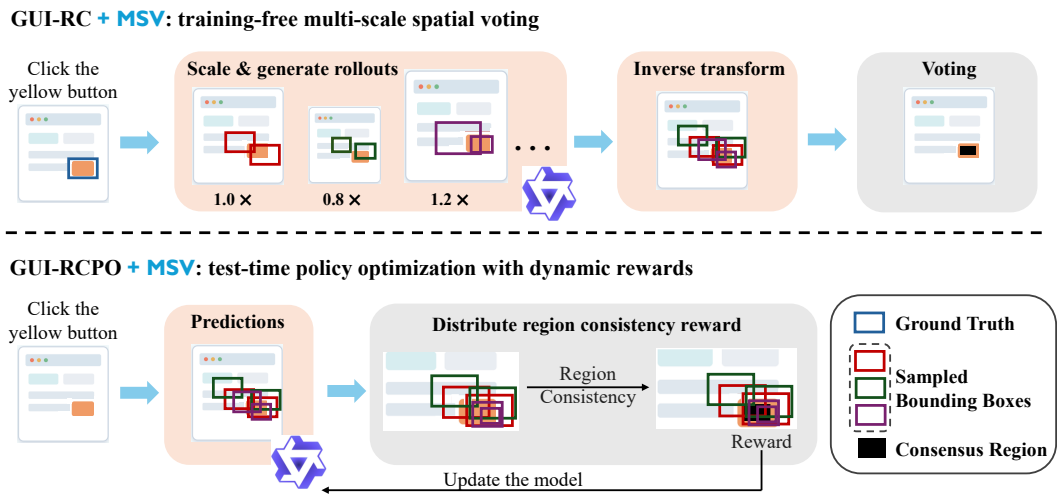

Figure 2: Illustration of the proposed MSV. The core idea is to apply multiple scales over the entire image before prediction rollouts, followed by spatial aggregation as in GUI-RC (Du et al., 2026). MSV also provides more precise rewards for test-time policy optimization.

leverages the unlabeled test dataset for policy optimization and then predicts on the same dataset using the updated model.

3.2 Multi-scale Spatial Voting

MSV generates multiple views by isotropically resizing the original screenshot I. For a scale factor s, the resized image is defined as

$$I^{(s)} = \text{Resize}\left(I, \lfloor sH \rfloor, \lfloor sW \rfloor\right), \tag{2}$$

where the height and width are scaled by the same factor, *i.e.*, $s_x = s_y = s$. The original screenshot is included as the view with $s = 1$, as illustrated in Figure 2.

Let (x, y) be a point in the original image coordinate system. Its coordinate in the resized image is $(x', y') = (sx, sy)$. Therefore, a prediction (x', y') obtained from the resized image is mapped back to the original image by $(x, y) = \left(\frac{x'}{s}, \frac{y'}{s}\right)$. Point-based predictions are first mapped back to the original coordinates and expanded into a square region of side length α:

$$r^{(s)} = \left[\frac{x'}{s} - \frac{\alpha}{2}, \frac{y'}{s} - \frac{\alpha}{2}, \frac{x'}{s} + \frac{\alpha}{2}, \frac{y'}{s} + \frac{\alpha}{2}\right]. \tag{3}$$

For region-based predictions, a bounding box $b = [x'_1, y'_1, x'_2, y'_2]$ in the resized image is transformed into $r^{(s)}$ as follows,

$$r^{(s)} = \left[\frac{x'_1}{s}, \frac{y'_1}{s}, \frac{x'_2}{s}, \frac{y'_2}{s}\right]. \tag{4}$$

All predictions from the original and resized views are then represented in the original image coordinate system. We follow previous work (Du et al., 2026) to construct a voting grid and extract the consensus region:

$$v(x, y) = \sum\nolimits_s \mathbb{I}\left[(x, y) \in r^{(s)}\right]. \tag{5}$$

The final prediction is obtained by selecting the connected component with the maximum vote value.

3.3 Integration with Off-the-Shelf Methods

We select four existing test-time GUI grounding methods and integrate the proposed MSV strategy with them. We describe the key points of each integration as follows.

- **GUI-RC** (Du et al., 2026) uses multiple grounding predictions from various rollouts and performs spatial voting to extract a consensus region. `MSV` extends this procedure by introducing multiple resized views of the input screenshot and mapping all predictions back to the original coordinate system before spatial voting. To ensure the same computation budget of K predictions, we distribute the K samples across the original view and N augmented views. The base number of samples for each augmented view is $q = \left\lfloor \frac{K}{N+1} \right\rfloor$, and the original image receives the remaining $r = K - q \times N$ samples.
- **RegionFocus** (Luo et al., 2025b) proposes focal points to extract targeted sub-regions, predicts actions individually for these sub-regions, and aggregates them via selection into a single refined action for standard inference. `MSV` also extends this by generating multiple resized views of the input screenshot, thereby producing more diverse candidates for final selection.
- **MVP** (Zhang et al., 2026) generates multiple views by cropping sub-regions from the original screenshot, and performs spatial clustering on all predicted coordinates to return the largest cluster's centroid. `MSV` also extends this by generating multiple resized views of the input screenshot, thereby producing more diverse candidates for final centroid calculation.
- **GUI-RCPO** (Du et al., 2026) extends GUI-RC and offers a training-based solution that enables test-time RL through region consistency signals as rewards for policy optimization. In particular, the region consistency reward for the k-th mapped prediction r_k is defined as

$$R_{\text{rc}}(k) = \frac{1}{|r_k| v_{\max}} \sum_{(x,y) \in r_k} v(x, y), \tag{6}$$

  where $|r_k|$ is the area of r_k, $v(x, y)$ is the voting value at position (x, y), and $v_{\max} = \max_{(x,y)} v(x, y)$. Predictions that overlap with high-consistency regions receive larger rewards, whereas spatially inconsistent predictions receive smaller rewards. As shown in Figure 2, `MSV` provides more precise regions and rewards for RL algorithms. Moreover, GUI-RCPO-RC applies GUI-RC on the updated model; `MSV` instead performs multi-scale voting.

4 Experiments

4.1 Setup

Models. We evaluate on three popular vision-language architectures, namely Qwen2.5-VL-Instruct (3B and 7B variants) (Bai et al., 2025) and GUI-G^2-7B (Tang et al., 2026). Note that GUI-G^2-7B is fine-tuned from Qwen2.5-VL-7B-Instruct within the VLM-R1 (Shen et al., 2025) framework. For simplicity, we use the same instruction template across different models (see Appendix A.1).

Datasets and metrics. We quantify grounding performance using standard Grounding Accuracy (Cheng et al., 2024), where a predicted coordinate, or the centroid of a predicted bounding box, is considered correct if it falls strictly within the target ground-truth box. We evaluate on four mainstream GUI benchmarks of varying complexity. ScreenSpot (Cheng et al., 2024) and ScreenSpot-v2 (Wu et al., 2025c) assess broad multi-platform capabilities across Mobile, Web, and Desktop interfaces, while ScreenSpot-Pro (Li et al., 2025) serves as a more challenging testbed with fine-grained, high-resolution professional application layouts. OSWorld-G (Xie et al., 2025) is a comprehensive benchmark covering diverse task types, including text matching (Text), element recognition (Element), layout understanding (Layout), and precise manipulation (Manipulation).

Implementation details. For all training-free and TTRL methods, each query image is augmented to three scales using factors of 0.8, 1.0 (original), and 1.2. As for GUI-RC (Du et al., 2026), we generate 64 candidate predictions per query, distributed as evenly as possible across the three scale views, with any remainder allocated to the original image. Sampling is performed with a temperature of 0.5, and the hyperparameter α is set to 50. For the TTRL variant, GUI-RCPO, we build upon the VLM-R1 (Shen et al., 2025) framework and conduct training without ground-truth annotations. Each instance uses the same multi-scale augmentation and 64-rollout scheme, but with a sampling temperature of 0.7. We optimize the model for 2 epochs with a global batch size of 64, a learning rate of 1×10^{-7}, and a KL penalty factor $\beta = 0.04$. All training and evaluation are conducted on 8 NVIDIA A100-80GB GPUs.

Table 1: Results of MSV combined with existing GUI grounding methods on the ScreenSpot dataset. MSV is highlighted in light blue, and changes in average performance are shown in green and red.

Model	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-3B-Instruct	95.97	68.56	86.60	55.71	80.43	63.59	77.12
RegionFocus (Luo et al., 2025b)	95.24	70.74	82.99	62.86	77.39	65.05	77.28
+ MSV	95.60	72.05	86.08	62.86	82.17	66.99	79.25 +1.97
GUI-RC (Du et al., 2026)	96.70	72.49	88.66	60.00	83.48	63.59	79.32
+ MSV	95.60	72.05	90.21	63.57	85.65	71.36	81.29 +1.97
MVP (Zhang et al., 2026)	75.09	46.72	81.44	59.29	75.65	65.05	67.69
+ MSV	89.38	66.81	87.11	59.29	78.26	66.02	75.86 +8.17
GUI-RCPO (Du et al., 2026)	96.70	77.29	93.30	67.14	87.39	72.82	83.88
+ MSV	96.70	77.29	94.85	66.43	87.83	75.73	84.59 +0.71
GUI-RCPO-RC (Du et al., 2026)	96.70	77.29	93.30	68.57	88.26	72.33	84.12
+ MSV	97.80	80.35	93.30	66.43	90.00	78.64	86.01 +1.89
Qwen2.5-VL-7B-Instruct	95.60	79.91	85.57	72.86	85.65	75.73	83.73
RegionFocus (Luo et al., 2025b)	93.41	76.42	88.66	66.43	86.09	71.36	81.76
+ MSV	95.97	78.17	92.27	72.86	88.26	73.30	84.59 +2.83
GUI-RC (Du et al., 2026)	96.34	81.22	91.75	72.86	88.70	77.18	85.85
+ MSV	96.70	82.97	92.27	77.14	89.13	76.70	86.79 +0.94
MVP (Zhang et al., 2026)	94.51	78.17	91.75	79.29	83.91	72.82	84.04
+ MSV	96.34	83.41	92.27	79.29	85.65	76.21	86.32 +2.28
GUI-RCPO (Du et al., 2026)	95.97	82.10	90.72	69.29	90.43	81.07	86.32
+ MSV	95.97	81.22	91.24	75.71	89.57	80.58	86.71 +0.39
GUI-RCPO-RC (Du et al., 2026)	95.97	82.97	89.18	70.71	90.00	81.55	86.40
+ MSV	96.70	82.97	94.85	79.29	90.43	80.58	88.29 +1.89
GUI-G^2-7B	96.70	89.96	94.33	84.29	90.87	87.38	91.19
RegionFocus (Luo et al., 2025b)	93.41	87.77	94.33	83.57	91.30	87.38	90.09
+ MSV	95.24	89.08	93.81	85.00	93.04	87.86	91.19 +1.10
GUI-RC (Du et al., 2026)	96.70	89.96	94.33	85.71	91.30	86.41	91.27
+ MSV	97.07	90.83	95.88	88.57	91.74	88.35	92.45 +1.18
MVP (Zhang et al., 2026)	95.60	86.46	95.36	90.72	90.87	85.44	90.72
+ MSV	96.70	88.21	95.88	90.00	93.48	86.89	92.14 +1.42
GUI-RCPO (Du et al., 2026)	96.34	90.83	94.33	88.57	89.13	87.86	91.95
+ MSV	96.70	90.83	94.33	86.43	91.30	89.32	91.98 +0.03
GUI-RCPO-RC (Du et al., 2026)	96.34	90.39	94.85	88.57	89.13	87.38	91.43
+ MSV	96.70	90.39	94.85	85.71	92.61	89.81	92.22 +0.79

Table 2: Results of MSV combined with existing grounding methods on the ScreenSpot-V2 dataset.

Model	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-3B-Instruct	97.67	74.87	87.78	58.73	86.36	65.40	80.27
RegionFocus (Luo et al., 2025b)	97.29	75.92	84.44	67.46	82.14	67.09	80.35
+ MSV	96.90	76.96	87.78	68.25	85.00	69.62	82.08 +1.73
GUI-RC (Du et al., 2026)	97.67	76.96	90.00	65.87	87.14	65.82	82.08
+ MSV	98.06	79.06	92.22	66.67	89.64	72.57	84.67 +2.59
MVP (Zhang et al., 2026)	77.13	52.36	80.00	63.49	76.43	67.93	70.60
+ MSV	89.53	71.73	88.33	61.90	80.36	69.62	78.22 +7.62
GUI-RCPO (Du et al., 2026)	99.22	81.68	95.00	73.81	92.14	73.42	87.11
+ MSV	99.22	81.15	95.56	72.22	92.86	75.11	87.42 +0.31
GUI-RCPO-RC (Du et al., 2026)	99.22	82.20	95.56	72.22	92.50	74.68	87.42
+ MSV	100.0	83.25	95.00	75.40	95.00	78.90	89.31 +1.89
Qwen2.5-VL-7B-Instruct	98.84	84.29	86.67	74.60	88.93	78.48	86.56
RegionFocus (Luo et al., 2025b)	95.74	81.15	86.11	64.29	87.86	72.15	82.94
+ MSV	98.45	84.82	95.56	74.60	89.29	78.06	87.81 +4.87
GUI-RC (Du et al., 2026)	99.22	85.86	91.67	73.81	91.43	80.59	88.44
+ MSV	99.22	86.91	95.56	76.98	92.86	80.17	89.70 +1.26
MVP (Zhang et al., 2026)	97.29	81.15	93.89	80.95	86.79	75.95	86.48
+ MSV	98.84	87.96	95.00	80.95	89.64	80.59	89.47 +2.99
GUI-RCPO (Du et al., 2026)	98.84	86.91	92.78	74.60	92.50	82.28	89.31
+ MSV	99.22	86.39	93.33	78.57	93.21	83.12	90.09 +0.78
GUI-RCPO-RC (Du et al., 2026)	98.84	86.39	92.22	76.98	92.50	82.28	89.39
+ MSV	99.22	88.48	95.56	80.95	94.64	86.50	91.90 +2.51
GUI-G^2-7B	99.22	93.19	97.22	85.71	94.64	86.92	93.40
RegionFocus (Luo et al., 2025b)	95.74	90.10	95.00	86.51	95.00	85.23	91.90
+ MSV	96.90	94.76	96.67	86.51	95.71	87.76	93.55 +1.65
GUI-RC (Du et al., 2026)	99.61	93.19	97.22	87.30	94.64	86.50	93.55
+ MSV	100.0	93.72	97.22	89.68	95.36	88.19	94.42 +0.87
MVP (Zhang et al., 2026)	97.67	90.05	97.22	91.27	94.29	86.08	92.92
+ MSV	99.61	91.10	97.22	92.06	95.71	87.34	94.10 +1.18
GUI-RCPO (Du et al., 2026)	98.84	92.67	96.11	86.51	93.93	86.50	92.92
+ MSV	98.84	92.67	96.67	88.10	94.64	87.34	93.47 +0.55
GUI-RCPO-RC (Du et al., 2026)	98.84	92.67	96.67	86.51	93.57	86.50	92.92
+ MSV	98.45	91.62	96.67	86.51	95.71	88.61	93.55 +0.63

Table 3: Results of MSV combined with existing grounding methods on the ScreenSpot-Pro dataset.

Model	Development		Creative		CAD		Scientific		Office		OS		Avg.
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-3B-Instruct	22.08	1.38	25.38	4.69	33.84	4.20	42.36	10.91	35.03	9.43	20.56	6.74	20.87
RegionFocus (Luo et al., 2025b)	37.66	6.21	38.07	6.25	35.86	5.59	42.36	10.00	45.76	24.53	28.04	8.99	27.13
+ MSV	40.26	5.52	40.61	6.25	46.97	4.20	45.83	13.64	54.80	16.98	24.30	10.11	30.04 +2.91
GUI-RC (Du et al., 2026)	24.68	3.45	27.41	7.81	35.35	9.09	45.14	10.91	45.20	18.87	21.50	6.74	24.10
+ MSV	31.17	4.83	29.95	6.25	43.43	6.99	51.39	13.64	50.85	18.87	25.23	11.24	27.83 +3.73
MVP (Zhang et al., 2026)	38.96	9.66	34.01	12.50	50.00	6.99	60.42	14.55	57.06	22.64	42.06	8.99	33.33
+ MSV	50.00	7.59	36.04	15.62	50.51	5.59	59.72	20.91	56.50	20.75	46.73	15.73	35.48 +2.15
GUI-RCPO (Du et al., 2026)	37.66	4.83	34.52	9.38	42.42	4.90	50.00	16.36	54.24	15.09	23.36	7.87	28.84
+ MSV	35.06	5.52	34.52	10.94	41.92	4.90	51.39	17.27	55.93	15.09	25.23	8.99	29.22 +0.38
GUI-RCPO-RC (Du et al., 2026)	36.36	6.21	35.03	9.38	41.41	6.29	47.92	18.18	57.06	18.87	20.56	8.99	29.16
+ MSV	42.86	6.90	37.56	9.38	47.98	4.90	59.72	17.27	60.45	18.87	32.71	12.36	33.27 +4.11
Qwen2.5-VL-7B-Instruct	31.17	2.76	13.71	3.12	21.72	8.39	40.97	6.36	41.81	15.09	33.64	11.24	20.87
RegionFocus (Luo et al., 2025b)	38.31	7.59	18.27	7.81	27.27	6.99	42.36	7.27	52.54	24.53	32.71	14.61	25.17
+ MSV	46.10	5.52	24.37	3.12	33.33	6.29	43.06	7.27	50.28	22.64	28.97	7.87	26.12 +0.95
GUI-RC (Du et al., 2026)	43.51	2.07	15.23	3.12	29.29	5.59	40.28	8.18	41.81	22.64	32.71	14.61	23.34
+ MSV	52.60	4.83	15.74	4.69	29.80	6.29	41.67	10.91	53.11	22.64	35.51	15.73	26.57 +3.23
MVP (Zhang et al., 2026)	62.34	21.38	50.76	21.88	60.10	17.48	70.83	23.64	75.71	35.85	63.55	30.34	48.13
+ MSV	70.13	22.76	51.27	15.62	57.58	17.48	61.11	23.64	74.58	33.96	67.29	31.46	47.75 −0.38
GUI-RCPO (Du et al., 2026)	48.05	4.14	19.80	3.12	29.80	8.39	42.36	11.82	46.33	13.21	41.12	12.36	24.93
+ MSV	49.35	2.76	18.78	6.25	27.27	9.09	39.58	11.82	47.46	15.09	35.51	11.24	25.17 +0.24
GUI-RCPO-RC (Du et al., 2026)	50.00	4.83	17.77	3.12	28.79	8.39	40.28	12.73	47.46	18.87	42.99	11.24	26.06
+ MSV	58.44	4.14	21.32	9.38	34.85	6.29	52.78	12.73	56.50	24.53	35.51	17.98	30.30 +4.24
GUI-G^2-7B	55.84	15.17	55.33	18.75	57.58	16.08	72.92	26.36	74.58	37.74	43.93	22.47	45.48
RegionFocus (Luo et al., 2025b)	68.18	24.83	57.36	15.62	60.10	13.99	77.78	29.09	74.15	45.28	51.40	21.35	49.21
+ MSV	75.97	26.21	61.42	20.31	62.63	18.18	78.47	30.00	79.10	50.94	52.34	25.84	52.56 +3.35
GUI-RC (Du et al., 2026)	57.79	15.86	52.28	18.75	56.57	13.99	75.69	28.18	73.45	39.62	43.93	19.10	45.16
+ MSV	74.03	28.97	62.44	21.88	64.14	17.48	79.86	30.00	81.92	43.40	49.53	28.09	53.07 +7.91
MVP (Zhang et al., 2026)	82.47	37.24	70.05	26.56	69.70	27.27	80.56	35.45	83.05	58.49	70.09	37.08	60.34
+ MSV	82.47	40.69	73.10	29.69	68.69	29.37	83.33	37.27	84.18	52.83	78.50	41.57	62.37 +2.03
GUI-RCPO (Du et al., 2026)	56.49	17.24	58.88	20.31	56.06	13.29	76.39	30.00	75.71	35.85	41.12	19.10	46.05
+ MSV	58.44	16.55	60.41	21.88	58.08	12.59	77.78	28.18	74.58	35.85	41.12	19.10	46.49 +0.44
GUI-RCPO-RC (Du et al., 2026)	45.45	14.48	54.82	21.88	41.41	11.89	65.97	28.18	62.15	32.08	36.45	14.61	39.09
+ MSV	70.78	29.66	65.48	26.56	62.12	19.58	80.56	30.00	80.23	41.51	48.60	30.34	53.19 +14.10

4.2 Main Results

Tables 1 to 4 evaluate MSV on four benchmarks by integrating it with five existing test-time grounding methods across three backbones.

On ScreenSpot and ScreenSpot-v2, MSV consistently improves performance across all backbones and baselines, with particularly evident gains when combined with MVP and RegionFocus. Even on the stronger GUI-G^2-7B model, MSV still yields positive gains, though the margins are smaller. On ScreenSpot-Pro, MSV delivers the largest improvements. Methods such as GUI-RC and GUI-RCPO-RC benefit substantially, which aligns with our motivation: when target elements vary drastically in size, single-scale predictions produce correlated errors, whereas multi-scale voting breaks this correlation and recovers correct localizations. A few combinations remain nearly unchanged, which is expected since some baselines already incorporate multi-view cropping and are therefore less sensitive to additional

Table 4: Results of MSV combined with existing grounding methods on the OSWorld-G dataset.

Model	Text	Element	Layout	Manipulation	Avg.
Qwen2.5-VL-3B-Instruct	40.00	32.03	31.80	14.39	27.25
RegionFocus (Luo et al., 2025b)	44.17	35.29	35.98	16.67	31.27
+ MSV	49.58	40.85	41.84	21.97	36.08 +4.81
GUI-RC (Du et al., 2026)	46.25	33.33	33.89	22.73	30.98
+ MSV	47.08	37.25	38.91	18.94	33.73 +2.75
MVP (Zhang et al., 2026)	50.42	37.25	41.00	21.97	35.29
+ MSV	54.17	38.89	41.84	23.48	36.67 +1.38
GUI-RCPO (Du et al., 2026)	51.67	35.95	37.66	21.21	34.31
+ MSV	53.75	37.91	39.33	21.97	36.47 +2.16
GUI-RCPO-RC (Du et al., 2026)	52.50	34.97	39.33	26.52	35.10
+ MSV	59.17	43.79	47.70	23.48	41.77 +6.67
Qwen2.5-VL-7B-Instruct	48.50	37.58	38.49	19.70	34.31
RegionFocus (Luo et al., 2025b)	49.17	42.81	44.77	19.70	37.84
+ MSV	51.67	44.77	47.28	22.73	40.00 +2.16
GUI-RC (Du et al., 2026)	48.33	38.56	39.75	18.18	34.90
+ MSV	59.17	40.85	44.35	25.00	39.22 +4.32
MVP (Zhang et al., 2026)	56.25	46.08	52.30	33.33	45.29
+ MSV	61.25	50.98	56.49	34.09	48.63 +3.34
GUI-RCPO (Du et al., 2026)	55.00	40.52	43.51	19.70	37.84
+ MSV	59.17	46.73	50.63	27.27	44.31 +6.47
GUI-RCPO-RC (Du et al., 2026)	55.42	40.52	43.93	21.97	38.43
+ MSV	63.33	47.39	52.30	31.06	45.88 +7.45
GUI-G^2-7B	70.42	54.58	59.83	40.15	54.31
RegionFocus (Luo et al., 2025b)	76.25	56.54	59.41	48.48	58.04
+ MSV	77.50	64.05	64.44	49.24	61.96 +3.92
GUI-RC (Du et al., 2026)	70.42	55.23	60.25	41.67	55.10
+ MSV	74.58	61.76	64.02	47.73	60.00 +4.90
MVP (Zhang et al., 2026)	75.00	63.73	64.44	49.24	61.57
+ MSV	73.75	63.07	66.53	44.70	59.80 −1.77
GUI-RCPO (Du et al., 2026)	73.75	56.86	60.25	44.70	56.86
+ MSV	75.00	58.50	58.58	46.21	57.84 +0.98
GUI-RCPO-RC (Du et al., 2026)	72.92	57.19	59.41	43.18	56.67
+ MSV	76.67	61.44	63.60	48.48	60.00 +3.33

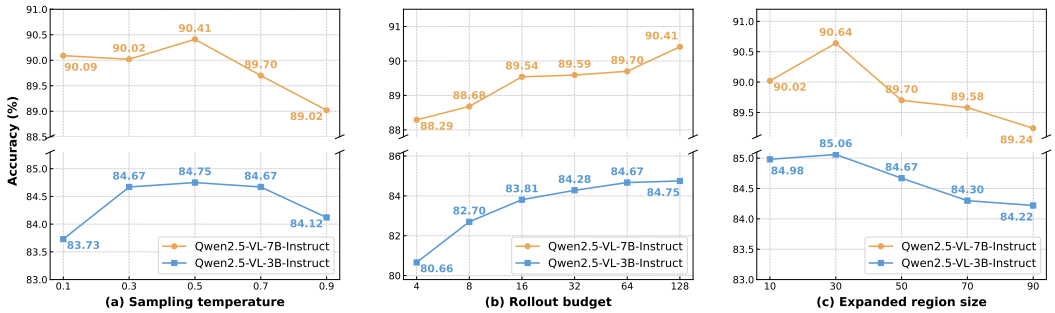

Figure 3: Results on ScreenSpot-V2 for MSV combined with GUI-RC (Du et al., 2026), where we vary the sampling temperature, the number of rollouts K, and the expanded region size α.

Table 5: Results on ScreenSpot-V2 for MSV combined with GUI-RC (Du et al., 2026), where we compare MSV with cropping and style augmentation strategies.

	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-7B-Instruct							
MSV	99.22	86.91	95.56	76.98	92.86	80.17	**89.70**
Cropping	98.84	85.86	91.11	73.81	91.43	78.48	87.89
Style Augmentation	98.22	85.96	93.19	78.47	92.21	80.43	89.08
Qwen2.5-VL-3B-Instruct							
MSV	98.06	79.06	92.22	66.67	89.64	72.57	**84.67**
Cropping	97.29	78.53	89.44	64.29	87.50	68.78	82.63
Style Augmentation	98.84	76.96	93.89	65.87	88.57	67.09	83.41

scale augmentation. On OSWorld-G, MSV also improves RegionFocus across all task types, including the most difficult precise manipulation category. This confirms that the benefit of MSV extends beyond simple text matching to layout understanding and fine-grained manipulation.

In summary, the results across all four benchmarks support two conclusions. First, scale variation is an important yet overlooked factor in test-time GUI grounding, especially on high-resolution and fine-grained tasks. Second, MSV is a simple and general strategy that can be plugged into existing methods without additional labels, improving both training-free and training-based grounding.

4.3 Analysis

On the ScreenSpot-v2 benchmark, we use GUI-RC (Du et al., 2026) to study model sensitivity to key inference hyperparameters (sampling temperature, rollout budget K, and expansion region size α), alternative augmentations, and different multi-scale augmentation configurations.

Sampling temperature. As shown in Figure 3(a), performance is highly sensitive to temperature. A moderate temperature around 0.5 optimally balances diversity and stability. Lower temperatures limit spatial exploration, while higher temperatures add noise that disrupts consensus voting.

Rollout budget K. Rollouts are distributed across original and scaled views. As shown in Figure 3(b), performance rises rapidly as K increases from 4 to 32 and plateaus around $K = 64$. For Qwen2.5-VL-3B, accuracy improves from 80.66% at $K = 4$ to 84.67% at $K = 64$. Further scaling to $K = 128$ yields diminishing returns, indicating that 64 samples suffice for reliable voting. Qwen2.5-VL-7 B shows a similar trend.

Expanded region size α. For point-based models, or region-based models with point predictions, α defines the simulated attention region. As shown in Figure 3(c), accuracy follows a bell-shaped curve, peaking between $\alpha = 50$ and $\alpha = 70$. Small α underestimates the target boundary, while large α includes irrelevant context, shifting the predicted center.

Table 6: Results on ScreenSpot-V2 for MSV combined with GUI-RC (Du et al., 2026), where we vary the scaling factors during square scaling and further consider non-square scaling.

	Qwen2.5-VL-3B-Instruct			Qwen2.5-VL-7B-Instruct		
Scaling Factors	Text Avg.	Icon Avg.	Avg.	Text Avg.	Icon Avg.	Avg.
Square Scaling						
1.2	93.04	72.93	84.28	96.24	80.80	89.53
0.8	93.03	71.66	83.73	95.54	81.23	89.31
1.2 0.8 (default)	93.31	73.47	84.67	95.82	81.77	89.70
1.1 0.9	92.76	72.74	84.04	95.96	81.77	89.78
1.3 0.7	94.01	72.56	84.67	95.26	82.49	89.70
1.4 0.6	93.73	72.75	84.59	96.38	82.85	90.49
1.5 0.5	93.73	71.84	84.20	96.10	83.21	90.49
1.1 1.2 0.8	92.76	74.55	84.83	95.82	83.39	90.41
1.1 1.2 0.9 0.8	93.87	74.37	85.38	95.54	82.67	89.94
1.1 1.2 1.3 0.9 0.8	92.90	74.73	84.98	95.40	83.75	90.33
1.1 1.2 1.3 0.9 0.8 0.7	93.46	74.55	85.22	95.96	82.16	89.94
1.1 1.2 1.3 1.4 0.9 0.8 0.7	93.59	74.91	85.46	96.24	83.40	90.64
1.1 1.2 1.3 1.4 0.9 0.8 0.7 0.6	93.73	75.45	85.77	96.38	82.67	90.41
Non-square Scaling						
1.2–0.8 1.2–0.8	91.78	72.20	83.25	95.68	79.78	88.76
0.8–1.2 0.8–1.2	93.03	70.76	83.33	94.98	79.60	88.29

Alternative augmentations. We compare MSV against *cropping augmentation*, which upsamples a local region but loses global context, and *style augmentation*, which applies pixel-level transformations while preserving geometry. As shown in Table 5, our multi-scale method achieves the best overall performance. Cropping augmentation yields the lowest accuracy, as the loss of global layout cues impairs spatial reasoning. Style augmentation remains competitive for text targets but degrades icon localization, since GUI icons rely on original color and style semantics. Overall, MSV best balances global context and visual feature preservation, providing the most robust pseudo-labels.

Augmentation strategies. As shown in Table 6, for Qwen2.5-VL-3B, a single scaled view is suboptimal (84.28% with a 1.2 factor and 83.73% with a 0.8 factor). Combining both up- and downscaling boosts accuracy to 84.67%, and using seven isotropic scaling factors reaches 85.77%, showing that finer scale diversity improves spatial invariance. Anisotropic resizing (different horizontal and vertical scaling) causes a noticeable drop, indicating that isotropic proportions are essential for preserving UI structural features. Similar observations are obtained for Qwen2.5-VL-7B.

Generalization performance. We further study the generalization performance of MSV when combined with the TTRL method GUI-RCPO (Du et al., 2026). Specifically, we update the model using unlabeled data from ScreenSpot-v2 and evaluate performance on ScreenSpot and ScreenSpot-Pro. Due to space limitations, we refer the reader to Appendix A.2 for the results. We find that MSV still boosts the performance of GUI-RCPO and its augmented variant, GUI-RCPO-RC.

5 Conclusion

In this paper, we identify scale variation as an overlooked factor in test-time GUI grounding and propose Multi-scale Spatial Voting (MSV), which resizes the full screenshot to multiple resolutions, samples predictions at each scale, and aggregates them by spatial voting. MSV keeps the full layout in view and avoids losing the target element. Experiments on four datasets with various backbones show that MSV consistently improves both training-free and training-based test-time grounding methods. It also yields gains when combined with test-time RL. These results suggest that scale is a simple but useful dimension for label-efficient GUI grounding. In the future, we will extend MSV to more existing test-time GUI grounding methods and validate its effectiveness during training the GUI models as well.

AI USE STATEMENT

Generative AI tools were used to polish the writing and improve the clarity of the manuscript. All AI-assisted edits were reviewed and revised by the authors, who take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

We provide implementation details and hyperparameter settings in the main paper. We are committed to open-sourcing the code upon acceptance.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Liangyu Chen, Hanzhang Zhou, Jianan Zhang, Panrong Tong, Xu Zhang, Quyu Kong, Chen Liu, Yuqi Liu, Wenxuan Wang, Yue Wang, et al. Ui-ins: Enhancing GUI grounding with multi-perspective instruction as reasoning. In *Proc. ICLR*, pp. 54418–54444, 2026.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing GUI grounding for advanced visual gui agents. In *Proc. ACL*, pp. 9313–9332, 2024.

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. In *Proc. AAAI*, pp. 30593–30601, 2026.

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 *Proc. ICLR*, pp. 30851–30883, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for GUI agents. In *Proc. CVPR*, pp. 14281–14290, 2024.

Weitai Kang, Bin Lei, Gaowen Liu, Caiwen Ding, and Yan Yan. Guirlvg: Incentivize GUI visual grounding via empirical exploration on reinforcement learning. In *Proc. ICLR*, pp. 63861–63875, 2026.

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.

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 *Proc. CVPR*, pp. 19498–19508, 2025.

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*, 2025.

Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, and Fei Wu. Infiguiagent: A multimodal generalist GUI agent with native reasoning and reflection. In *Proc. EACL*, pp. 1035–1051, 2026.

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 *Proc. AAAI*, pp. 17608–17616, 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*, 2025a.

Tiange Luo, Lajanugen Logeswaran, Justin Johnson, and Honglak Lee. Visual test-time scaling for GUI agent grounding. In *Proc. ICCV*, pp. 19989–19998, 2025b.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proc. EMNLP*, pp. 20286–20332, 2025.

Anthony Nguyen. Improved GUI grounding via iterative narrowing. *arXiv preprint arXiv:2411.13591*, 2024.

Rui Qian, Xin Yin, Chuanhang Deng, Zhiyuan Peng, Jian Xiong, Wei Zhai, and Dejing Dou. Uground: Towards unified visual grounding with unrolled transformers. In *Proc. ICML*, 2026.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, et al. GUI-G^2: Gaussian reward modeling for GUI grounding. In *Proc. AAAI*, pp. 33214–33222, 2026.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, et al. GUI agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*, 2024b.

Wenkai Wang, Xiyun Li, Hongcan Guo, Wenhao Yu, Tianqing Fang, Haitao Mi, Dong Yu, and Shengyu Zhang. Measure twice, click once: Co-evolving proposer and visual critic via reinforcement learning for GUI grounding. In *Proc. ACL*, pp. 20964–20984, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proc. ICLR*, 2023.

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. In *Proc. EMNLP*, pp. 26246–26256, 2025a.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. GUI-Actor: Coordinate-free visual grounding for GUI agents. In *Proc. NeurIPS*, pp. 15101–15128, 2025b.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: Foundation action model for generalist GUI agents. In *Proc. ICLR*, pp. 5090–5108, 2025c.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, et al. Scaling computer-use grounding via user interface decomposition and synthesis. In *Proc. NeurIPS*, 2025.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous GUI interaction. In *Proc. ICML*, pp. 69772–69805, 2025.

Shiyu Xuan and Zechao Li. Test-time self-evolving GUI visual grounding via reflection-guided on-policy self-distillation. *arXiv preprint arXiv:2608.11191*, 2026.

Dong Yan, Jian Liang, Yanbo Wang, Shuo Lu, Ran He, and Tieniu Tan. What if consensus lies? selective-complementary reinforcement learning at test time. In *Proc. ACL*, pp. 28957–28970, 2026.

Yongcan Yu, Lingxiao He, Jian Liang, Kuangpu Guo, Meng Wang, Qianlong Xie, Xingxing Wang, and Ran He. Understanding and mitigating spurious signal amplification in test-time reinforcement learning for math reasoning. In *Proc. ACL Findings*, pp. 37424–37436, 2026.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, et al. Se-gui: Enhancing visual grounding for GUI agents via self-evolutionary reinforcement learning. In *Proc. NeurIPS*, pp. 127658–127679, 2025.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.

Yunzhu Zhang, Zeyu Pan, Zhengwen Zeng, Shuheng Shen, Changhua Meng, and Linchao Zhu. Mvp: Multiple view prediction improves GUI grounding. In *Proc. CVPR*, pp. 27482–27492, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. In *Proc. NeurIPS*, pp. 131459–131483, 2025.

A Appendix

A.1 Instruction Prompt Template

Throughout this paper, we employ the following template: `You are a GUI analysis expert. Based on the screenshot, your task is to locate the UI element that best matches the instruction: '{instruction}'. You MUST output exactly one bounding box in the format [x1, y1, x2, y2] strictly. Do not include any explanations, descriptions, or additional text. **Output format: [x1, y1, x2, y2]**.` The `instruction` placeholder is replaced with the actual input at inference time.

A.2 Generalization performance

We also provide results for models trained on ScreenSpot-v2 and tested on ScreenSpot and ScreenSpot-Pro using GUI-RCPO, to further validate the effectiveness of the proposed `MSV` strategy. Performance is shown in Tables 7 and 8, respectively.

A.3 Training dynamics

As shown in Figure 4, we track the evaluation accuracy of Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct across training steps to analyze the learning dynamics of `MSV`. Both models exhibit a steep performance increase within the first 30 to 50 steps across all benchmarks, indicating that our multi-scale voting mechanism provides dense, high-quality pseudo-labels for rapid policy optimization. In later stages, the trajectories diverge due to model capacity differences. The 7B model

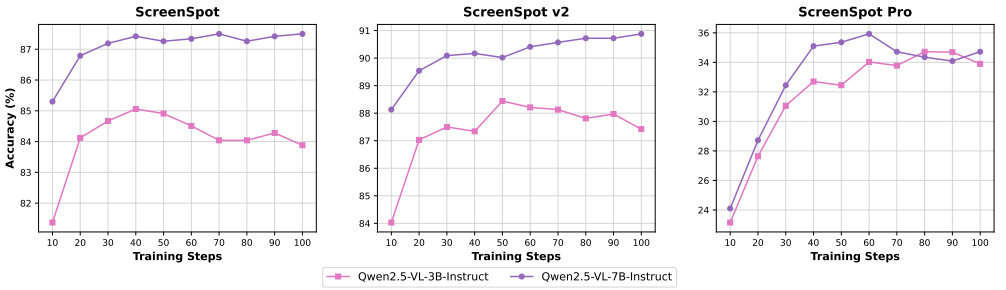

Figure 4: Results on ScreenSpot-V2 for MSV combined with GUI-RCPO (Du et al., 2026), where we vary training steps.

remains stable and maintains a high plateau even on the challenging out-of-distribution ScreenSpot-Pro benchmark, whereas the 3B model peaks earlier and shows a slight degradation. This suggests that while our method effectively bootstraps grounding capabilities, higher-capacity models are better equipped to leverage spatial consensus rewards and generalize without overfitting to the training distribution.

Table 7: Generalization Results on ScreenSpot.

Model	Mobile		Desktop		Web		Avg.
	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-3B-Instruct	95.97	68.56	86.60	55.71	80.43	63.59	77.12
GUI-RCPO (Du et al., 2026)	96.70	77.73	93.30	67.14	87.83	72.33	83.96
+ MSV	96.70	76.86	94.85	66.43	88.26	74.27	84.36$_{+0.40}$
GUI-RCPO-RC (Du et al., 2026)	96.70	77.29	93.81	70.71	87.83	71.84	84.28
+ MSV	97.80	78.60	92.78	67.86	90.00	76.70	85.46$_{+1.18}$
Qwen2.5-VL-7B-Instruct	95.60	79.91	85.57	72.86	85.65	75.73	83.73
GUI-RCPO (Du et al., 2026)	95.97	82.10	90.72	70.00	90.43	82.01	86.40
+ MSV	95.97	82.97	91.75	72.86	90.87	82.04	87.26$_{+0.86}$
GUI-RCPO-RC (Du et al., 2026)	96.34	81.66	90.21	71.43	90.00	82.04	86.56
+ MSV	95.24	83.41	93.81	79.29	91.30	82.04	88.29$_{+1.73}$
GUI-G^2-7B	96.70	89.96	94.33	84.29	90.87	87.38	91.19
GUI-RCPO (Du et al., 2026)	95.79	89.52	92.78	85.71	90.43	86.89	90.72
+ MSV	96.34	89.96	94.33	86.43	91.74	88.83	91.75$_{+1.03}$
GUI-RCPO-RC (Du et al., 2026)	96.34	89.52	94.33	85.71	90.43	86.89	91.04
+ MSV	96.34	89.08	94.33	87.14	93.48	90.78	92.30$_{+1.26}$

Table 8: Generalization Results on ScreenSpot-Pro.

Model	Development		Creative		CAD		Scientific		Office		OS		Avg.
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Qwen2.5-VL-3B-Instruct	22.08	1.38	25.38	4.69	33.84	4.20	42.36	10.91	35.03	9.43	20.56	6.74	20.87
GUI-RCPO (Du et al., 2026)	38.96	4.83	35.03	10.94	42.42	4.20	49.31	17.27	54.80	15.09	23.36	7.87	29.10
+ MSV	37.36	5.14	33.49	11.94	44.43	7.29	51.00	16.45	56.37	16.09	25.30	8.87	29.84 +0.74
GUI-RCPO-RC (Du et al., 2026)	37.01	4.83	35.53	9.38	40.91	6.29	51.39	19.09	55.93	16.98	22.43	8.99	29.41
+ MSV	42.86	6.90	37.56	14.06	45.45	6.29	56.52	16.36	62.15	18.87	30.84	12.36	32.95 +3.54
Qwen2.5-VL-7B-Instruct	31.17	2.76	13.71	3.12	21.72	8.39	40.97	6.36	41.81	15.09	33.64	11.24	20.87
GUI-RCPO (Du et al., 2026)	47.40	3.45	19.29	3.12	30.30	8.39	41.67	11.82	45.76	13.21	41.12	12.36	25.68
+ MSV	49.35	2.76	20.30	6.25	27.78	8.39	40.97	10.91	46.33	16.98	35.51	11.24	25.36 −0.32
GUI-RCPO-RC (Du et al., 2026)	49.35	4.83	16.75	1.56	29.29	6.99	40.28	12.73	47.46	18.87	41.12	13.48	25.74
+ MSV	57.79	8.97	21.83	6.25	29.29	10.49	45.83	15.45	59.89	18.87	41.12	14.61	30.23 +4.49
GUI-G^2-7B	55.84	15.17	55.33	18.75	57.58	16.08	72.92	26.36	74.58	37.74	43.93	22.47	45.48
GUI-RCPO (Du et al., 2026)	55.84	14.48	53.30	18.75	54.04	11.89	72.22	29.09	72.88	35.85	35.51	19.10	43.45
+ MSV	57.14	14.48	52.79	18.75	55.56	13.99	75.00	25.45	74.01	32.08	19.10	39.25	44.15 +0.70
GUI-RCPO-RC (Du et al., 2026)	55.19	14.08	52.28	18.75	54.04	13.99	75.00	30.00	75.14	35.85	35.51	20.22	44.09
+ MSV	68.83	26.90	54.82	20.31	58.59	20.28	76.39	28.18	74.58	39.62	40.19	21.35	48.51 +4.42