

Zoom in, Click out: Unlocking and Evaluating the Potential of Zooming for GUI Grounding

Anonymous ACL submission

Abstract

Grounding is a fundamental capability for building graphical user interface (GUI) agents. Although existing approaches rely on large-scale bounding box supervision, they still face various challenges, such as cross-platform generalization, complex layout analysis, and fine-grained element localization. In this paper, we investigate **zoom** as a strong yet underexplored prior for GUI grounding, and propose a training-free method, **ZoomClick**. By characterizing four key properties of zoom (i.e., pre-zoom, depth, shrink size, minimal crop size), we unlock its full capabilities for dynamic spatial focusing and adaptive context switching. Experiments demonstrate that our method significantly boosts the performance of both general vision-language and specialized GUI grounding models, achieving state-of-the-art results on several mainstream benchmarks, e.g., UI-Venus-72B attains a 73.1% success rate on ScreenSpot-Pro. Besides, our method achieves a better trade-off between performance, computation, and cost, enabling smaller baselines to outperform much larger ones while reducing GFLOPs and tokens cost. Furthermore, we present **GUIZOOM-BENCH**, a benchmark for evaluating model’s adaptability to zoom, aiming to inspire future research on improving zoom for further training and test-time scaling in GUI grounding tasks.

1 Introduction

As digital platforms (e.g., desktops, webpages, and applications) increasingly serve as the primary medium for human-computer interaction (HCI), there has been growing interest in developing GUI agents (Liu et al., 2025; Lu et al., 2024; OpenAI et al., 2024b; Xie et al., 2025), which understands user instructions, perceives screenshots, plans operation steps, and performs interactive actions. A fundamental capability of GUI agent perception is grounding (Cheng et al., 2024; Li et al., 2025a; Wu

et al., 2024), mapping the textual descriptions of a target visual element to its on-screen coordinates, thereby generating bounding boxes for localization.

Existing approaches mainly leverage general multimodal large language models (MLLMs) (Bai et al., 2025; Wang et al., 2024; Yang et al., 2025a) or visual grounding models to finetune specialized GUI grounding models (Gu et al., 2025; Qin et al., 2025), and further introduce several post-training (Zhao et al., 2025) or test-time scaling (Luo et al., 2025; Xu et al., 2024) strategies to enhance their performance (Ye et al., 2025). Among these strategies, **zoom** (Nguyen, 2025) is intuitively appealing, since grounding requires not only understanding the global context (Nayak et al., 2025) but also distinguishing fine-grained local regions (Li et al., 2025a), and zoom enables the model to dynamically switch between them.

However, the implementation of zoom in existing studies remains limited. For example, they either broadly aggregate predictions from a single zoom applied to multiple ROIs (Luo et al., 2025; Wu et al., 2025a), or recursively zoom into a specific region without an effective stopping criterion, which may result in missed small icons or produce excessively high-resolution crops that are out-of-distribution (OOD). Although certain studies optimize zoom via training, such as introducing IoU-aware losses to refine zoom ranges (Park et al., 2025) or maintaining global-local consistency via zoom (Lee et al., 2025), these approaches are still auxiliary techniques, as the intrinsic test-time scaling (Kaya et al., 2025; Byun et al., 2025) potential of zoom itself is largely underexplored.

In this paper, we explore several highly practical insights by conducting a comprehensive empirical analysis of zoom.

Insight 1: Zoom is simple yet effective, its full capability better activates the vision-language and grounding priors of foundation models. We propose **ZoomClick**, a training-free method that in-

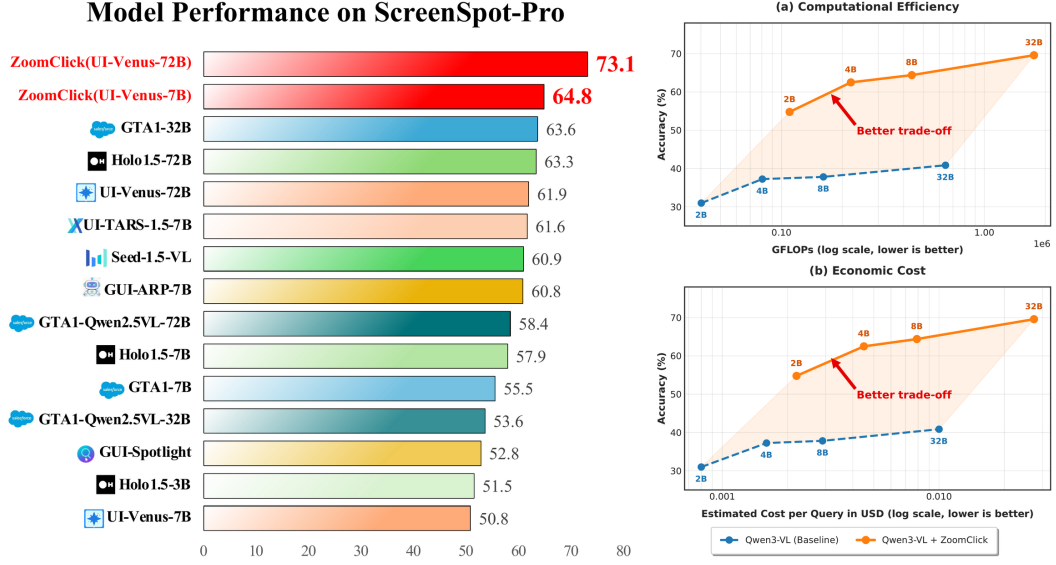


Figure 1: Left: Performance of existing GUI grounding methods on the ScreenSpot-Pro(Li et al., 2025a) benchmark, where GTA1-32B(Yang et al., 2025b) serves as the previous state-of-the-art method. Right: Accuracy-efficiency trade-offs w/o ZOOMCLICK across model scales.

tegrates four critical properties of zoom (i.e., pre-zoom, depth, shrink size, and minimal crop size). Specifically, ZoomClick undergoes a three-stage pipeline.

When to zoom in. ZoomClick begins with a Pre-Zoom that compares a global prediction with four local predictions over a predefined grid. A local candidate becomes the starting point when its distance to the global candidate falls below a threshold, ensuring early correction with full context still available and steering the model toward regions that receive high confidence both globally and locally.

How to zoom. Each iteration crops a region of the specified shrink size directly in the original coordinate system, preventing iterative drift from relative cropping and avoiding boundary overflow that introduces irrelevant context.

When to click out. Termination occurs when no target is detected, the zoom depth is reached, or the crop hits the minimal size, enabling resolution-adaptive stopping and preventing over-zooming into unrecognizable context.

On multiple mainstream benchmarks, ZoomClick enables state-of-the-art (SOTA) models to achieve substantial breakthroughs, and even allows smaller models to surpass their previously larger counterparts.

Insight 2: Zoom is far from saturated, but current benchmarks are unable to expose its essential shortcomings. We present **GUIZOOM-BENCH**, which collects evaluation data based on

two principles—the zoom depth UI-Venus-72B first hit the target and whether a prediction maintains after subsequent zooming. We samples into five types: easy-normal, hard-normal, easy-mislead, hard-mislead and hard-est. Our results further reveals the unresolved headroom of zoom in complex layouts, fine-grained elements, and resolution-mismatch scenarios, offering interpretable standards to design more robust and generalizable zoom methods.

Overall, our contributions can be summarized as follows:

- A training-free GUI grounding method that integrates four important capabilities of zoom (i.e., pre-zoom, depth, shrink size, and minimal crop size) within a three-stage pipeline.
- A benchmark for evaluating the zoom capabilities of different GUI grounding models, delivering a more specific and interpretable quantitative basis for future zoom improvements.
- Experiments demonstrate that ZoomClick enables existing models to achieve state-of-the-art performance in GUI grounding tasks, with smaller models attaining results comparable to those of larger models, e.g., UI-Venus-72B attains a 73.1% success rate on ScreenSpot-Pro and UI-Venus-7B with ZoomClick outperforms original UI-Venus 72B by 2.2%.

2 Related Works

2.1 GUI Agents

GUI agent autonomously selects actions to accomplish user-specified tasks (Chen et al., 2024; Ding et al., 2023; Yin et al., 2024; Zhou et al., 2024) by perceiving the interactive interface. Early approaches rely on structured text extracted from GUI environments, such as HTML (Ding et al., 2023; Lu et al., 2024; Zhou et al., 2024) or accessibility trees, and utilize large language models (LLMs) to process them (Liu et al., 2023). To better understand visual rendering and layout variability, recent approaches have shifted toward MLLMs (Guo et al., 2025) to directly operate on screenshots. Specifically, modular approaches adopt vision-language models (VLMs) as high-level semantic planners (Luo et al., 2025; Wu et al., 2025a; Yang et al., 2025b), complemented by specialized grounding models. In contrast, end-to-end approaches jointly train planning and localization capabilities within a unified framework. Post-training methods are further introduced to enhance reasoning and adaptability, e.g. reinforcement learning (RL) (Tang et al., 2025; Zhou et al., 2025b).

2.2 GUI Grounding

GUI grounding maps natural language instructions to on-screen visual elements (Wu et al., 2025b), serving as a prerequisite for the planning and action of GUI agents. Existing approaches typically formulate the task as coordinate prediction and finetune VLMs (Gu et al., 2025; Lee et al., 2025; Qin et al., 2025; Park et al., 2025) or general grounding models to leverage their perceptual priors. This paradigm has led to the development of specialized GUI grounding models (Zhou et al., 2025a), large-scale screenshot-text-position paired datasets (Cheng et al., 2024; Li et al., 2025a; Xie et al., 2024) and corresponding collection/synthesis pipelines, as well as standardized evaluation benchmarks. Notably, zoom has demonstrated strong effectiveness in many visual grounding tasks (Shen et al., 2025; Wu and Xie, 2023), and recent works have attempted to introduce this mechanism into above models. R-VLM (Park et al., 2025) leverages a two-step coarse-to-fine zoom in, followed by an IoU-aware weighted loss to ensure high IoU grounding. DiMo-GUI (Wu et al., 2025a) decouples layouts into texts and icons, maintaining a search path for each modality, and select between the two

candidates based on model judgement. ReGUIDE (Lee et al., 2025) adopts a statistics-driven strategy, using Kernel Density Estimation to determine the optimal crop region for robust majority voting at test time. Other methods such as GUI-Cursor (Zhao et al., 2025) and RegionFocus (Luo et al., 2025) maintain a single search path with history actions, and relies on the model to determine the next action. These methods leverage zoom either as an auxiliary tool for training or as a simple rule-based heuristic at inference time. As a result, its full potential remains underdeveloped, and a notable performance gap persists between these practices and what zoom can really achieve. Our work close this gap through a comprehensive analysis of zoom behaviors, revealing the principles required to fully unlock its capability.

3 Zoom in, Click out : A Practical Paradigm for Unlocking the Potential of Zoom-in Grounding

We introduce ZOOMCLICK, a training-free test-time search strategy. ZOOMCLICK uses a fixed shrink ratio together with a minimum crop size and simple boundary handling, which preserves context and stabilizes narrowing without per-model parameter adjustment. This design works consistently across different grounding models and leads to more reliable multi-step localization.

3.1 Problem Setup

Given a high-resolution GUI image $I \in \mathbb{R}^{H \times W \times 3}$ and a language query q , a grounding model $\mathcal{G}$ outputs a normalized point $\hat{p} = (\hat{x}, \hat{y}) \in [0, 1]^2$ with respect to its input view. We maintain a viewport $V = (v_x^1, v_y^1, v_x^2, v_y^2) \in [0, 1]^4$ that specifies a region of I in normalized coordinates. The predicted point in the original image coordinates is

$$p_r = (v_x^1 + (v_x^2 - v_x^1)\hat{x}, v_y^1 + (v_y^2 - v_y^1)\hat{y}), \quad (1)$$

and the corresponding pixel location is

$$p_{px} = (\text{round}(W \cdot p_r^x), \text{round}(H \cdot p_r^y)). \quad (2)$$

We start with the full-image viewport $V_0 = (0, 0, 1, 1)$ and iteratively update the viewport during zooming; all updates operate on the same mapping above.

3.2 ZoomClick

3.2.1 When to Zoom in

According to our studies in Fig 3b, great first-step accuracy leads to massive performance boost in

Method	Training-Free	Pre-Zoom (Sec.3.2.1)	Multi-Step Zoom (Sec. 3.2.2)	Plain Input	Plain Output	Adaptive Context Retention (Sec. 3.2.2)
GUI-SPOTLIGHT(Lei et al., 2025)	✗	✗	✓	✗	✗	✗
GUI-Cursor(Zhao et al., 2025)	✗	✗	✗	✗	✗	✗
R-VLM	✓*	✗	✓	✓	✗	✗
ReGUIDE(Lee et al., 2025)	✗	✗*	✗	✓	✓	✗
DiMo-GUI(Wu et al., 2025a)	✓	✗	✓	✓	✓	✗
GMS(Li et al., 2025b)	✓	✗	✓	✗	✗	✗
RegionFocus(Luo et al., 2025)	✓	✗	✓	✗	✗	✗
Ours (ZoomClick)	✓	✓	✓	✓	✓	✓

Table 1: **Feature comparison of recent GUI grounding works.** *Plain Input* denotes that no additional visual context beyond the raw screenshot is provided; *Plain Output* denotes that no VLM-as-judge or external reranking module is used. * in the second column means that R-VLM can be deployed without training at inference time. * in the third column means that ReGUIDE take global-local consistency into consideration, but different from prezoom.

latter steps. Therefore, we introduce **Pre-Zoom**, a single-shot agreement test between the full image and K non-overlapping patches in the first iteration, as an effective way to *ensure a reliable start*. Let

$$p^{\text{dir}} = \mathcal{G}(I, q), \quad p^{(k)} = \mathcal{G}(I^{(k)}, q),$$

denote the predictions on the full image and the k -th patch, respectively, each mapped back to the original image coordinates. We compute pixel-space distances

$$d_k = \|p^{\text{dir}} - p^{(k)}\|_2, \quad k^* = \arg \min_k d_k,$$

and use a threshold τ (in pixels) to select the initial point:

$$p^{(1)} = \begin{cases} p^{(k^*)}, & d_{k^*} < \tau, \\ p^{\text{dir}}, & \text{otherwise.} \end{cases}$$

Intuitively, if any tile closely agrees with the global prediction, it provides a *cleaner local context* for initiating zoom; otherwise, we retain the global view to avoid premature narrowing.

3.2.2 How to Zoom in

Instead of a one-shot crop, ZOOMCLICK progressively navigates and refines the viewport around each prediction. This gradual shrinking enables the model to iteratively correct localization errors and converge to the precise target.

At each iteration t , ZOOMCLICK updates a localized view by shrinking the current viewport around the predicted point. Given viewport size (W_t, H_t) and prediction $p^{(t)}$, we compute a new crop size:

$$(\tilde{W}, \tilde{H}) = (\max(\lfloor \rho W_t \rfloor, m), \max(\lfloor \rho H_t \rfloor, m)),$$

where $\rho \in (0, 1)$ is a fixed shrink ratio and m is a minimum crop size that preserves essential context.

When to Click out The minimum crop size m acts as a lower bound that prevents the view from

collapsing into an overly tight and noisy region. Formally, the effective zoom level at iteration t satisfies:

$$\min(W_t, H_t) \geq m,$$

ensuring that a non-trivial spatial neighborhood around $p^{(t)}$ is always preserved. This *context floor* keeps the view within the model’s training resolution regime and retains semantic cues (e.g., surrounding text, icon grouping, layout) needed for reliable grounding.

Viewport Boundary Handling If a crop exceeds the image boundary, we apply: shift (move window inside), clip (trim overflow), or shrink (reduce window while keeping $p^{(t)}$ centered). We use shift by default; clip is useful when the prediction is already accurate and further motion is unnecessary.

Termination We stop after T iterations or once the crop reaches the minimum size. The final click is the last mapped prediction $p_r^{(T)}$, converted to pixel coordinates p_{px} .

4 GUIZOOM-BENCH: Benchmarking the Behavioral Patterns of Zoom-in Strategies

The potential of zoom is far from saturated, yet existing GUI grounding benchmarks fail to reveal its essential shortcomings or the model-specific limitations that emerge under different zoom conditions. Current evaluations do not clarify when zoom improves localization, when it induces harmful context loss or distraction, nor why certain models collapse under deeper zoom levels—leaving little guidance for advancing zoom-aware methods. To address this gap, we introduce **GUIZOOM-BENCH**, a benchmark designed to systematically dissect zoom behavior, exposing both the failure modes of zoom itself and the adaptability boundaries of different model families. These insights not

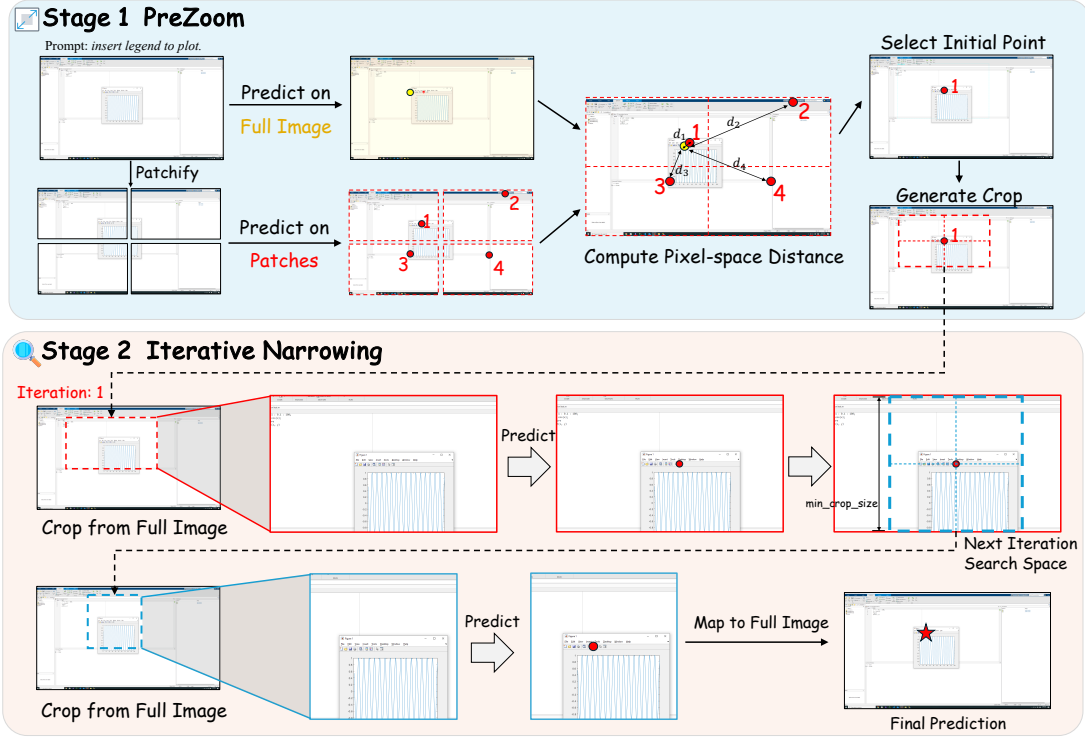
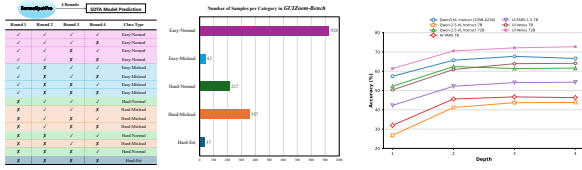


Figure 2: **Model Framework of ZoomClick.** Min_crop_size (3.2.2) represents the lower bound of viewport during Iterative Narrowing.



(a) Data organization in our proposed GUIZOOM-BENCH. (b) Accuracy vs. zoom depth across models.

Figure 3: **Overview of benchmark design and zoom-depth evaluation.** (a) Data organization of GUIZOOM-BENCH. (b) Model accuracy as zoom depth increases.

only pinpoint the true sources of performance gains but also provide concrete directions for developing more effective and robust zoom-aware grounding models.

4.1 Five Zoom Behaviors: Difficulty × Reliability

Zoom does not benefit all samples in the same way. Some targets are easily recognized without zooming, others gradually emerge through progressive narrowing, and some are even misled by lost context or visual distractors. **GUIZOOM-BENCH** captures these behavioral differences by partitioning samples into five subsets that reveal *how* zoom shapes model perception—highlighting when it aids, when it distracts, and where it reaches its limits.

- **easy-normal.** The target is visually salient and contextually clear; In this case, *excessive zooming can hurt* by unnecessarily discarding useful context.
- **easy-mislead.** The model starts correct but becomes wrong after zooming, as distractors grow more salient once global cues vanish. This reflects a *loss of stability* caused by over-focusing.
- **hard-normal.** The target is small or visually subtle, initially hard to locate but gradually revealed through zooming. This showcases the *productive role* of zoom in exposing fine-grained details.
- **hard-mislead.** The scene contains distractors visually similar to the target, making zooming prone to shift attention away. These cases capture *context-induced confusion* under narrowing.
- **hard-est.** Even repeated zooming fails to clarify the target due to extreme subtlety or vague instruction. This subset defines the *upper limit* of zoom-enabled grounding.

Together, these five categories delineate a two-dimensional space of **difficulty × reliability**, where

difficulty captures the inherent visual and contextual complexity of a target, and *reliability* reflects the model’s behavioral stability under zooming.

4.2 Collection Method and Criteria

We construct **GUIZOOM-BENCH** by reorganizing samples from *ScreenSpot-Pro*, a large-scale GUI grounding dataset with high-resolution interfaces and dense, fine-grained targets—making it a *natural testbed* for analyzing zoom behavior.

To capture how zoom affects grounding dynamics, we run our state-of-the-art ZOOMCLICK model (UI-Venus-72B) under a fixed four-step zoom-in procedure and record correctness at each iteration. Each ScreenSpot-Pro sample is evaluated under four zoom-in rounds using the SOTA grounding model, producing a correctness sequence $\{s_1, s_2, s_3, s_4\}$, where $s_t=1$ indicates a correct click at iteration t .

From these correctness sequences, we extract two behavioral factors that define the taxonomy shown in Fig. 3a (Left). The *iteration of first correctness* quantifies the sample’s **difficulty**—how many zoom steps are needed before the model finds the target—while the *stability of correctness* reflects **reliability**—whether the model correctness remains once zoomed in.

By crossing these two dimensions, we categorize every sample into one of five behavioral types (*easy-normal*, *easy-mislead*, *hard-normal*, *hard-mislead*, *hard-est*). The number of samples per category is shown in Fig. 3a (Right).

5 Experiments

5.1 Experimental Setup

Model Selection We evaluate ZOOMCLICK on two strong GUI grounding models with automatic localization: *Qwen3-VL* (Yang et al., 2025a) (general-purpose VLM) and *UI-Venus* (GUI-specific multimodal agent). Together, they span open-domain reasoning and specialized UI interaction, allowing us to test ZOOMCLICK’s generalization across paradigms under a unified zoom inference schedule.

Datasets and Benchmarks We evaluate on two complementary benchmarks for realistic desktop-scale GUIs. *ScreenSpot-Pro* provides high-resolution professional interfaces with densely packed fine-grained elements, ideal for testing visual precision under zoom. *UI-Vision* (Nayak et al., 2025) includes diverse real-world software anno-

tated with elements and interaction traces, emphasizing functional understanding and realistic interaction contexts. Together, they combine visual complexity and interaction diversity, forming a natural testbed for assessing the robustness and effectiveness of zoom-based grounding.

Metric We report grounding accuracy—the proportion of correctly localized targets—under different zoom behaviors and datasets, following the official evaluation setup of each benchmark to ensure fair comparison. Full implementation details can be found in Appendix A.

5.2 Main Results

5.2.1 ScreenSpot-Pro

ZOOMCLICK yields *consistent and significant improvements* of grounding accuracy over base models. As shown in Table 2, ZOOMCLICK + Qwen3-VL-32B achieves a **34.4%** relative improvement over the base model, Qwen3-VL-32B. A similar trend holds for GUI-specific models: ZOOMCLICK + UI-Venus-7B achieves 30.6% relative accuracy gain, surpassing even the base UI-Venus-72B model. Moreover, even large and strong backbones benefit from ZOOMCLICK, with UI-Venus-72B gaining an 19.1% accuracy improvement, setting a new record of **73.1%** for ScreenSpot-Pro.

5.2.2 UI-Vision

As shown in Table 4, ZOOMCLICK delivers a substantial improvement over prior work on UI-Vision. When applied to UI-Venus-72B, it raises click accuracy from **25.5%** to **42.5%** (+66.7% relative), establishing a new state of the art. These gains persist across both *model scales* (7B–72B) and *model types* (UI-specific vs. general-purpose VLMs), demonstrating that our training-free refinement reliably enhances grounding across architectures rather than relying on model or dataset specialization.

5.2.3 Performance between Zoom-based Methods

As shown in Table 3, our *training-free* method achieves an overall accuracy of **44.0%** with *Qwen-2.5-VL-7B*, substantially outperforming the previous training-free method RegionFocus (32.1%) and the finetuned method GUI-Spotlight (38.7%), and closely matching ReGUIDE, which relies on additional training, demonstrating that carefully designed zoom strategies alone can already recover a

Methods	Development			Creative			CAD			Scientific			Office			OS			Avg		
	text	icon	avg	text	icon	avg	text	icon	avg	text	icon	avg	text	icon	avg	text	icon	avg	text	icon	avg
Proprietary Methods																					
GPT-4o(OpenAI et al., 2024a)	1.3	0.0	0.7	1.0	0.0	0.6	2.0	0.0	1.5	2.1	0.0	1.2	1.1	0.0	0.9	0.0	0.0	0.0	1.3	0.0	0.8
Claude-3.7-Sonnet(Anthropic, 2025)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	27.7
Seed-1.5-VL(Guo et al., 2025)	-	-	53.8	-	-	59.2	-	-	59.0	-	-	61.4	-	-	74.8	-	-	60.2	-	-	60.9
UI-TARS-1.5(Qin et al., 2025)	-	-	63.9	-	-	50.4	-	-	58.2	-	-	69.3	-	-	79.6	-	-	51.0	-	-	61.6
Open-Sourced Methods																					
OS-Atlas-7B(Wu et al., 2024)	33.1	1.4	17.7	28.8	2.8	17.9	12.2	4.7	10.3	37.5	7.3	24.4	33.9	5.7	27.4	27.1	4.5	16.8	28.1	4.0	18.9
Qwen2.5-VL-3B(Bai et al., 2025)	38.3	3.4	21.4	40.9	4.9	25.8	22.3	6.3	18.4	44.4	10.0	29.5	48.0	17.0	40.9	33.6	4.5	20.4	37.8	6.6	25.9
Qwen2.5-VL-7B	51.9	4.8	29.1	36.9	8.4	24.9	17.8	1.6	13.8	48.6	8.2	31.1	53.7	18.9	45.7	34.6	7.9	22.4	39.9	7.6	26.8
UGround-7B(Gou et al., 2025)	-	-	35.5	-	-	27.8	-	-	13.5	-	-	38.8	-	-	48.8	-	-	26.1	-	-	31.1
UGround-72B	-	-	31.1	-	-	35.8	-	-	13.8	-	-	50.0	-	-	51.3	-	-	25.5	-	-	34.5
UI-TARS-7B	58.4	12.4	36.1	50.0	9.1	32.8	20.8	9.4	18.0	63.9	31.8	50.0	63.3	20.8	53.5	30.8	16.9	24.5	47.8	16.2	35.7
UI-TARS-72B	63.0	17.3	40.8	57.1	15.4	39.6	18.8	12.5	17.2	64.6	20.9	45.7	63.3	26.4	54.8	42.1	15.7	30.1	50.9	17.5	38.1
Jedi-7B(Xie et al., 2025)	42.9	11.0	27.4	50.0	11.9	34.0	38.0	14.1	32.2	72.9	25.5	52.4	75.1	47.2	68.7	33.6	16.9	26.0	52.6	18.2	39.5
Qwen2.5-VL-32B	74.0	21.4	48.5	61.1	13.3	41.1	38.1	15.6	32.6	78.5	29.1	57.1	76.3	37.7	67.4	55.1	27.0	42.3	63.2	22.5	47.6
GTA1-7B(Yang et al., 2025b)	53.3	17.2	44.5	66.9	20.7	44.0	62.6	18.2	44.4	76.4	31.8	57.1	82.5	50.9	75.2	48.6	25.9	38.3	65.5	25.2	50.1
GTA1-32B	83.1	37.9	61.2	72.2	25.9	52.8	70.1	31.3	60.5	84.7	39.1	65.0	89.3	64.2	83.5	76.6	51.7	65.3	78.9	38.9	63.6
Ours																					
Qwen3-VL-32B + ZoomClick	74.0	13.1	44.5	74.2	23.1	52.8	60.4	23.4	51.3	78.5	30.9	57.9	81.9	41.5	72.6	66.4	24.7	47.4	72.6	24.0	54.0
Δ	88.3	55.9	72.6	82.8	38.5	64.2	80.7	42.2	71.3	90.2	40.9	68.9	93.8	77.4	90.0	84.1	51.7	69.4	86.5	48.8	72.1
	+14.3	+42.8	+28.1	+8.6	+15.4	+11.4	+20.3	+18.8	+20.0	+11.7	+10.0	+11.0	+11.9	+35.9	+17.4	+17.7	+27.0	+22.0	+13.9	+24.8	+18.1
UI-Venus-7B + ZoomClick	73.7	24.1	49.8	61.6	14.7	41.9	58.9	18.8	49.0	76.4	33.6	57.9	75.1	43.4	67.8	49.5	22.5	37.2	66.3	24.5	50.3
Δ	83.1	45.5	64.9	74.2	30.8	56.0	79.2	32.8	67.8	86.8	42.7	67.7	88.7	52.8	80.4	78.5	39.3	60.7	81.6	39.9	65.7
	+9.1	+21.4	+15.1	+12.6	+16.1	+14.1	+20.3	+14.0	+18.8	+10.4	+9.1	+9.8	+13.6	+9.4	+12.6	+29.0	+16.8	+23.5	+15.3	+15.4	+15.4
UI-Venus-72B + ZoomClick	82.5	33.8	58.9	73.7	30.8	55.7	65.5	29.7	56.7	84.0	42.7	66.1	82.5	58.5	77.0	73.8	36.0	56.6	76.6	36.8	61.4
Δ	88.3	51.0	70.2	81.8	49.7	68.3	85.3	48.4	76.2	91.0	48.1	72.4	91.5	66.0	85.7	79.4	52.8	67.3	86.4	51.5	73.1
	+5.8	+17.2	+11.3	+8.1	+18.9	+12.6	+19.8	+18.7	+19.5	+7.0	+5.4	+6.3	+9.0	+7.5	+8.7	+5.6	+16.8	+10.7	+9.8	+14.7	+11.7

Table 2: Comparison of Methods on ScreenSpot-Pro.

Method	Training-Free	Dev	Creative	CAD	Scientific	Office	OS	Ovr
GUI-Spotlight	$\times$	29.8	29.1	39.2	39.8	63.9	24.5	38.7
GUI-Cursor	$\times$	68.9	42.7	46.7	61.4	74.8	50.0	56.5
ReGUIDE	$\times$	-	-	-	-	-	-	44.4
RegionFocus	$\checkmark$	29.1	27.0	22.2	37.0	51.7	29.6	32.1
Ours	$\checkmark$	42.1	39.6	32.2	43.7	61.3	50.0	44.0

Table 3: Comparison of zoom-based grounding methods with Qwen2.5-VL-7B. $\checkmark$ indicates methods that require task-specific training, while $\times$ denotes training-free approaches.

large portion of the performance gap without any parameter updates.

Analysis The effectiveness of ZOOMCLICK stems from its simplicity: a training-free procedure that leverages the strong priors already embedded in modern VLMs and grounding-specific models. These models are generally capable of identifying semantically plausible regions, but often lack the mechanisms to translate coarse alignment into precise localization. ZOOMCLICK exposes and amplifies this through a minimal multi-step zooming routine, without introducing additional learning or architectural changes. Despite its simplicity, this lightweight refinement reliably converts existing semantic priors into state-of-the-art grounding performance on high-resolution desktop interfaces.

5.3 Ablations

Our ablation results reveal a clear and surprisingly consistent pattern: *implementing ZOOMCLICK with the simplest variant of each component achieves best performance*. These findings underscore a central insight of this work—successful zoom-based grounding comes from simply allowing the model’s inherent localization priors to operate without disruption.

Method	Basic Overall	Func Overall	Spatial Overall	Final Avg
Closed-Source VLMs				
GPT-4o	1.6	1.5	1.0	1.4
Gemini-1.5-pro(Team et al., 2024)	0.8	0.3	0.6	0.6
Claude-3.7-Sonnet	9.5	7.7	7.6	8.3
Open-Source VLMs				
Qwen-2.5VL-7B	1.2	0.8	0.5	0.9
InternVL2.5-8B(Chen et al., 2025)	2.5	2.8	1.0	2.1
Qwen-2VL-7B(Wang et al., 2024)	3.4	3.2	1.5	2.7
MiniCPM-V-8B(Yao et al., 2024)	7.1	5.3	1.5	4.3
Open-Source GUI Agents				
ShowUI-2B(Lin et al., 2024)	8.1	7.7	2.1	5.9
AriaUI-25.3B(Yang et al., 2025c)	12.2	14.0	4.0	10.1
UGround-v1-7B	15.4	17.1	6.3	12.9
OSAtlas-7B	12.2	11.2	3.7	9.0
UGround-7B	11.5	12.2	2.8	8.8
Aguvis-7B(Xu et al., 2025)	17.8	18.3	5.1	13.7
UI-TARS-7B	20.1	24.3	8.4	17.6
CogAgent-9B(Hong et al., 2024)	12.0	12.2	2.6	8.9
SeeClick-9.6B(Cheng et al., 2024)	9.4	4.7	2.1	5.4
UGround-v1-72B	27.9	26.7	14.9	23.2
UI-TARS-72B	31.4	30.5	14.7	25.5
Ours				
UI-Venus-7B	34.0	30.7	10.8	25.1
+ ZoomClick	43.1	39.9	19.1	34.0
Δ	+9.1	+9.2	+8.3	+8.9
Qwen3-VL-32B	44.2	42.2	24.4	36.6
+ ZoomClick	45.1	45.1	28.3	39.2
Δ	+0.9	+2.9	+3.9	+2.6
UI-Venus-72B	45.6	42.3	23.7	36.8
+ ZoomClick	50.2	46.6	31.7	42.5
Δ	+4.6	+4.3	+8.0	+5.7

Table 4: Comparison of Methods on UI-Vision.

5.3.1 When to Zoom in

The importance of Pre-zoom. As shown in Fig 3b, it’s progressively harder for the model to recover from earlier localization errors, demonstrating the importance of a reliable start. The results from Table 8a proves the effectiveness of Pre-zoom: across all models, Pre-zoom improves grounding accuracy at all depths—for instance, Qwen3-VL-32B rose from **67.2%**→**71.3%** (**+4.1%**) at Depth 2—indicating that early coarse localization provides a stable initialization for subsequent zooming. *Distance-based Pre-zoom yields better performance.* We evaluate two basic approach for Pre-zoom: VLM-based and Distance-based. As shown in Table 8b, the *Distance-based* pre-zoom yields consistently higher accuracy—surpassing VLM-based approach on Qwen3-VL-32B by **9.4%** and

on UI-Venus-72B by 5.1%.

Insights Findings above suggest a simple but actionable principle for future zoom-based grounding: *intervene only where necessary, and intervene minimally*. A small, well-placed adjustment to the initial region already produces the most meaningful gains, while extra semantic or heuristic reasoning adds little value and can introduce instability. This highlights a clearer direction for future work—before adding external complexity to the method, make full use of the intrinsic potential of zoom itself.

5.3.2 How to Zoom in

The effectiveness of Multi-step Zooming. As shown in Table 6, performing two consecutive $\times \frac{1}{2}$ zooms consistently outperforms a single $\times \frac{1}{4}$ shrink across all domains, notably, improving overall accuracy from **62.1%** to **63.9%** (+1.8%). Crucially, this self-correction arises from the model’s *inherent spatial priors* rather than explicit contextual injection—a trend validated by our controlled comparisons.

5.3.3 When to Click out

The effectiveness of minimum crop size. According to Table 7, two types of models exhibit markedly different sensitivities to `min_crop_size`: Qwen3-VL-32B shows a progressive accuracy increase as the minimum crop size grows, whereas UI-Venus-7B reaches its peak at smaller crop sizes and degrades consistently as `min_crop_size` increases.

Analysis The contrasting trends between general-purpose VLMs and UI-specialized models stem from their different *adaptability to zoom*. General VLMs benefit from larger crops because their reasoning depends heavily on global context, whereas UI-specialized models—optimized for small-window icon and widget recognition—perform best with tightly focused crops and quickly lose discriminative power as more background enters the view. This also explains the *mislead* phenomenon in Table 5, where models that rely on localized icon cues are more easily drawn toward visually salient but irrelevant regions. These behaviors validate our benchmark taxonomy by revealing each model family’s intrinsic strengths and weaknesses under zoom. They further point to a promising direction for future work: grounding models should incorporate **multi-resolution or multi-scale training** to better handle zoom-induced context changes and make the zoom pipeline itself more robust.

5.4 GUIZOOM-BENCH Results (Table 5)

(1) *Validation of benchmark construction.* Across model families, the curves consistently reveal the core challenges of zoom: most models surge in early depths (e.g., Qwen3-VL-32B improves from **56.1%**→**92.0%** on *easy-normal*) yet lose stability once the window becomes too narrow. In contrast, UI-specialized models—though strong on icon recognition—remain highly vulnerable to distractors, as reflected by their steady drop on *hard-mislead* (e.g., UI-Venus-7B falls from **35.1%**→**29.7%** across d1→d4). These confirm that it captures both shared pitfalls of zoom-based grounding and model-specific weaknesses.

(2) *Differences in upper-bound potential across models.* The varying gain profiles across depths also reveal clear differences in model capacity. General-purpose VLMs profit more from iterative narrowing (e.g., Qwen3-VL-32B gains **+36.0%** on *hard-normal* from d1→d2), whereas UI-centric models plateau earlier with much smaller deltas, reflecting strong local sensitivity but weaker global reasoning. These disparities indicate substantial headroom for model design depending on whether robustness or precision is the priority.

(3) *Implications for future training strategies.* The results suggest that next-generation grounding models should adopt *zoom-aware training*—e.g. multi-resolution supervision or dynamic cropping—to better handle scale changes. Improving robustness across context scales is essential for raising the performance ceiling of zoom-and-click pipelines and reducing mislead-related failures.

6 Conclusion

In this work, we revisit **zoom** as a structured mechanism that shapes how models perceive high-resolution GUIs, rather than auxiliary post-processing tools. We introduce ZOOMCLICK, a simple yet practical zoom paradigm that unlocks a model’s inherent spatial priors to progressively reveal fine-grained structure, achieving improved accuracy with reduced overhead. Complementing this, GUIZOOM-BENCH provides a behavior-based evaluation framework that characterizes how different models respond to zoom—highlighting where it helps, where it destabilizes predictions, and how these patterns relate to difficulty and reliability. Together, ZOOMCLICK and GUIZOOM-BENCH offer both a practical recipe for applying zoom and a diagnostic lens for understanding the dynamics behind zoom-enabled grounding.

7 Limitations

Our method is dictated by the model’s spatial and semantic priors, limiting the performance ceiling with the model’s built-in strengths and weaknesses. GUIZOOM-BENCH captures only desktop-scale conditions, that does not directly generalize to mobile interfaces or multi-step agent interaction workflows.

References

- Anthropic. 2025. Claude 3.7 sonnet and claude code. <https://www.anthropic.com/news/claude-3-7-sonnet>. Accessed: 2025-11-14.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao 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, and 8 others. 2025. *Qwen2.5-vl technical report*. *Preprint*, arXiv:2502.13923.
- Ji Young Byun, Young-Jin Park, Navid Azizan, and Rama Chellappa. 2025. *Test-time-scaling for zero-shot diagnosis with visual-language reasoning*. *Preprint*, arXiv:2506.11166.
- Qiang Chen, Ziqi Lin, and 1 others. 2024. *Cohtml-agent: A multimodal agent for web environments*. *arXiv preprint arXiv:2402.15288*.
- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. *Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling*. *Preprint*, arXiv:2412.05271.
- Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. *Seeclick: Harnessing gui grounding for advanced visual gui agents*. *Preprint*, arXiv:2401.10935.
- Yi Ding, Xiao Sun, Rui Zhang, Krisztian Balog, Yfatah He, and 1 others. 2023. *Mind2web: Towards a generalist web agent*. In *NeurIPS*.
- Boyuan Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025. *Navigating the digital world as humans do: Universal visual grounding for gui agents*. *Preprint*, arXiv:2410.05243.
- Zhangxuan Gu, Zhengwen Zeng, Zhenyu Xu, Xingran Zhou, Shuheng Shen, Yunfei Liu, Beitong Zhou, Changhua Meng, Tianyu Xia, Weizhi Chen, Yue Wen, Jingya Dou, Fei Tang, Jinzhen Lin, Yulin Liu, Zhenlin Guo, Yichen Gong, Heng Jia, Changlong Gao, and 5 others. 2025. *Ui-venus technical report: Building high-performance ui agents with rft*. *Preprint*, arXiv:2508.10833.
- Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, Jingji Chen, Jingjia Huang, Kang Lei, Liping Yuan, Lishu Luo, Pengfei Liu, Qinghao Ye, Rui Qian, Shen Yan, and 178 others. 2025. *Seed1.5-vl technical report*. *Preprint*, arXiv:2505.07062.
- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. *Cogagent: A visual language model for gui agents*. *Preprint*, arXiv:2312.08914.
- Mehmet Onurcan Kaya, Desmond Elliott, and Dim P. Papadopoulos. 2025. *Efficient test-time scaling for small vision-language models*. *Preprint*, arXiv:2510.03574.
- Hyunseok Lee, Jeonghoon Kim, Beomjun Kim, Jihoon Tack, Chansong Jo, Jaehong Lee, Cheonbok Park, Sookyo In, Jinwoo Shin, and Kang Min Yoo. 2025. *Reguide: Data efficient gui grounding via spatial reasoning and search*. *Preprint*, arXiv:2505.15259.
- Bin Lei, Nuo Xu, Ali Payani, Mingyi Hong, Chunhua Liao, Yu Cao, and Caiwen Ding. 2025. *GUI-SPOTLIGHT: Adaptive iterative focus refinement for enhanced gui visual grounding*. *Preprint*, arXiv:2510.04039.
- Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025a. *Screenspot-pro: Gui grounding for professional high-resolution computer use*. *Preprint*, arXiv:2504.07981.
- Zhecheng Li, Guoxian Song, Yiwei Wang, Zhen Xiong, Junsong Yuan, and Yujun Cai. 2025b. *Generalist scanner meets specialist locator: A synergistic coarse-to-fine framework for robust gui grounding*. *Preprint*, arXiv:2509.24133.
- Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. *Showui: One vision-language-action model for gui visual agent*. *Preprint*, arXiv:2411.17465.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. *Visual instruction tuning*. *Preprint*, arXiv:2304.08485.
- Zikang Liu, Junyi Li, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji rong Wen. 2025. *Pal-ui: Planning with active look-back for vision-based gui agents*. *Preprint*, arXiv:2510.00413.
- Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024. *Omniparser for pure vision based gui agent*. *Preprint*, arXiv:2408.00203.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Silvio Savarese, Caiming Xiong, and Junnan Li. 2025b. [Gta1: Gui test-time scaling agent](#). *Preprint*, arXiv:2507.05791.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025c. [Aria-ui: Visual grounding for gui instructions](#). *Preprint*, arXiv:2412.16256.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. [Minicpm-v: A gpt-4v level mllm on your phone](#). *Preprint*, arXiv:2408.01800.

Xianhang Ye, Yiqing Li, Wei Dai, Miancan Liu, Ziyuan Chen, Zhangye Han, Hongbo Min, Jinkui Ren, Xiantao Zhang, Wen Yang, and Zhi Jin. 2025. [Gui-arp: Enhancing grounding with adaptive region perception for gui agents](#). *Preprint*, arXiv:2509.15532.

Ke Yin, Shunyu Yao, and 1 others. 2024. [Browser-use: Teaching large language models to use a browser](#). *arXiv preprint arXiv:2402.17143*.

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. 2025. [Learning gui grounding with spatial reasoning from visual feedback](#). *Preprint*, arXiv:2509.21552.

Shijie Zhou, Viet Dac Lai, Hao Tan, Jihyung Kil, Wanrong Zhu, Changyou Chen, and Ruiyi Zhang. 2025a. [Gui-aima: Aligning intrinsic multimodal attention with a context anchor for gui grounding](#). *Preprint*, arXiv:2511.00810.

Wangchunshu Zhou, Xiang Xu, Sheng Shen, and 1 others. 2024. [Webarena: A realistic web environment for building autonomous agents](#). In *ICLR*.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. 2025b. [Gui-g1: Understanding rl-zero-like training for visual grounding in gui agents](#). *Preprint*, arXiv:2505.15810.

A Implementation Details

We adopt a fixed zoom depth of $T=3$ iterations with a shrink ratio of $\rho=0.5$, enforcing a minimum crop size of $m=768$ pixels to preserve essential context. The initial pre-zoom step employs a non-overlapping 2×2 patch grid to establish a reliable starting region. For results in main experiment, we use the clip boundary mode to maintain spatial alignment of localized regions. Unless specified, we use the shift boundary mode in ablations to ensure a fair comparison across models.

Center handling during zoom. When the zoom window centered at the predicted click would cross image boundaries, we control its behavior via a `center_mode` flag in the cropping function. Concretely, `shift` (default) keeps the target window size fixed and translates the window back into the valid image region whenever it overflows. `clip` keeps the predicted center fixed and simply intersects the window with the image bounds, so the effective crop may be smaller but the center is unchanged. `shrink` also fixes the center, but adaptively reduces the window width/height to the largest size that fits inside the image, which can lead to very small fields of view for points near the image borders.

Ablations on Pre-Zoom Strategy. For ablations in 5.3.1, we implement a *VLM-based* variant: we replace the heuristic distance selection with an explicit vision-language reasoning procedure. Specifically, we first invoke the VLM to perform a global grounding prediction on the original screenshot, producing one candidate location. Next, the same screenshot is evenly partitioned into a 2×2 grid, and the VLM is independently queried on each of the four patches to obtain four additional local predictions. This results in five candidate points in total. All five candidates are then projected back onto the original screenshot and visualized using identical circular markers (red filled circles) with uniform size, each annotated with a unique numerical index. A final VLM query is issued over this annotated image, prompting Qwen3-VL-32B to output the index corresponding to the point it considers most likely to represent the correct target location for the first-round grounding. The selected point is then used as the pre-zoom reference for subsequent cropping. Our default *distance-based* pre-zoom uses the same five candidates but simply chooses the patch-center nearest to the global

prediction.

Ablations on Context Injection. For ablations on context information, we select UI-TARS-1.5-7B as our base model. We design five default context-retention settings, all sharing the same zoom and cropping strategy, but differing only in how information from the previous round is conveyed to the current round:

1. *Prompt Only (Positive)*. The relative position of the previous click within the current cropped region is explicitly described in the prompt, accompanied by positive guidance encouraging the model to refine and correct its previous prediction.
2. *Prompt Only (Neutral)*. The relative position of the previous click is provided in the prompt as plain information, without any evaluative or guiding language, merely indicating that it corresponds to the last-round result.
3. *Visual Marking*. The previous click is annotated directly on the current image using a red cross marker, serving as a visual cue without textual explanation.
4. *Full Context Injection*. Both the previous round’s input image and its corresponding click point are supplied as auxiliary context alongside the current image, enabling the model to reason jointly over cross-round visual information.
5. *No Context*. No information from the previous round is provided; each round is performed independently.

B More Ablations

The Effectiveness of Context. Table 9 shows that the highest overall accuracy is achieved when no contextual information is injected across zoom rounds (54.1%), while all context-aware variants lead to performance degradation, most notably the Full Context Injection setting (19.0%). Among them, Visual Marking (Red Cross) exhibits the least drop, indicating that purely visual cues introduce less disturbance than prompt-based or multi-modal context, whereas textual descriptions of previous predictions consistently bias the model away from optimal visual reasoning. These results highlight a core limitation of training-free zoom-based grounding: externally introduced context shifts the

model’s implicit spatial distribution and often anchors it to earlier errors, leading to error accumulation during iterative refinement. Crucially, this does not suggest that context is inherently harmful, but rather that the model lacks the learned capacity to interpret such information when injected only at inference time. This motivates *treating context as a trainable guidance signal instead of an ad-hoc heuristic*, enabling future models to explicitly learn how to leverage historical localization traces as structured auxiliary information to improve robustness and iterative correction.

The Effectiveness of Boundary Handling Mode.

Across all three models in Table 10, shrink consistently yields the lowest accuracy (e.g., 63.6% vs. 72.1% for Qwen3-VL-32B), while shift and clip remain close. This is consistent with their geometric behavior: both shift and clip preserve a reasonably large field of view—either by translating the fixed-size crop back into the image (shift) or by intersecting it with the image bounds (clip). In contrast, shrink keeps the predicted center fixed and adaptively reduces the crop to the largest window that fits within the image. For points near the image borders (which are common in GUI layouts, e.g., toolbars and sidebars), this produces very small crops that discard most contextual information. As a result, subsequent iterations are forced to refine within an overly narrow and potentially misaligned region, amplifying early localization errors instead of correcting them. This systematic loss of context explains why shrink is consistently the worst-performing strategy.

The Selection of Distance Threshold. The distance reported here denotes the Euclidean pixel displacement between two consecutive predicted click locations produced in adjacent zoom rounds, measured in the original image space. Consecutive pairs are categorized as 1–1 (correct pairs) when both predictions fall inside the ground-truth bounding box, and as 0–1 / 1–0 (error pairs) when only one of them is correct, indicating unstable refinement. As shown in the table, correct pairs exhibit tightly clustered distances near zero (median < 10 px), reflecting smooth local refinement, whereas error pairs show substantially larger displacements (median often > 50 px) with heavy tails, corresponding to abrupt spatial jumps. The optimal thresholds across individual stages lie within a narrow range (33.4–52.2 px), and the aggregated distribution yields an optimal threshold of 50.7 px with

Model	easy-normal				easy-mislead				hard-normal				hard-mislead				hard-est			
	d1	d2	d3	d4	d1	d2	d3	d4	d1	d2	d3	d4	d1	d2	d3	d4	d1	d2	d3	d4
<i>Close-sourced Models</i>																				
GPT-5	2.2	9.5	14.8	13.7	3.1	2.4	2.4	2.4	5.5	5.5	6.5	6.9	0.0	8.1	2.7	5.4	0.7	2.8	4.8	4.2
Claude-4.5-Sonnet	30.7	—	—	—	5.9	—	—	—	4.9	—	—	—	10.5	—	—	—	6.7	—	—	—
<i>Open-Sourced Models</i>																				
Qwen3-VL-8B	54.2	89.0	88.5	87.9	26.2	38.1	47.6	42.9	24.9	55.3	55.3	56.7	16.2	32.4	40.5	45.9	8.1	19.9	18.8	18.2
Qwen3-VL-32B	56.1	92.0	92.2	92.0	23.8	35.7	45.2	47.6	30.4	69.6	67.3	67.3	21.6	45.9	45.9	54.1	10.1	27.2	25.5	25.5
Qwen2.5-VL-7B	41.1	61.2	63.5	63.3	7.1	9.5	7.1	9.5	12.9	21.7	28.1	28.6	0.0	10.8	16.2	13.5	3.1	8.1	9.0	9.5
UI-TARS-7B	48.5	66.1	66.1	64.8	16.7	19.0	21.4	19.0	11.5	30.4	30.4	32.3	5.4	10.8	24.3	18.9	6.2	8.4	11.8	12.9
UI-TARS-1.5-7B	62.8	74.0	77.0	77.3	19.0	26.2	28.6	28.6	21.8	35.0	37.3	39.2	18.9	24.3	21.6	18.9	6.5	12.0	11.2	10.4
UI-Venus-7B	73.7	84.1	86.6	86.4	31.0	38.1	40.5	42.9	28.1	47.0	54.8	57.6	18.9	35.1	35.1	29.7	8.7	14.6	16.0	16.2

Table 5: Comparison of Models on GUIZOOM-BENCH.

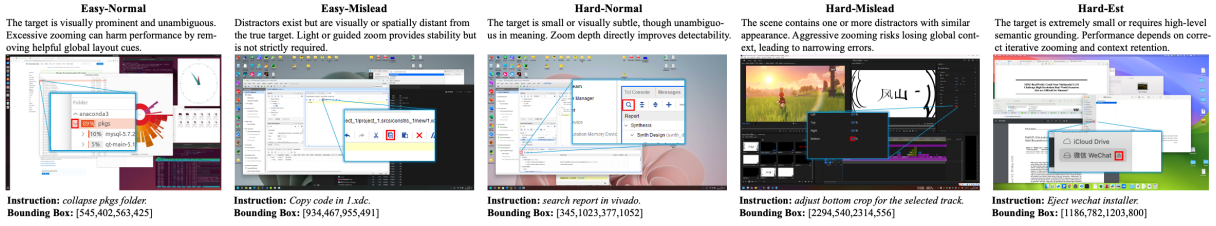


Figure 4: Data examples of each category in GUIZOOM-BENCH.

Strategy	Overall	Dev	Creative	CAD	Scientific	Office	OS
1/4 (one-step)	62.1	59.5	51.9	63.2	65.0	77.4	61.2
1/2 + 1/2 (two-step)	63.9	62.9	54.0	64.4	64.6	80.4	61.7

Table 6: Comparison of Zoom Strategies. Accuracy (%) under two zoom configurations: Two-step $\times \frac{1}{2}$ vs. One-step $\times \frac{1}{4}$. The two-step strategy performs two consecutive shrinks that each reduce the viewport to $\frac{1}{2}$ of its previous size, whereas the one-step strategy directly shrinks the viewport to $\frac{1}{4}$ of the original size in a single operation. The two-step strategy consistently outperforms the one-step alternative across all domains.

Model	0	384	512	640	768	896	1024
Qwen3-VL-32B	65.2	69.6	70.4	71.0	72.1	72.2	72.3
UI-Venus-7B	65.5	65.7	65.5	65.6	63.9	62.9	63.3

Table 7: Ablations on min_crop_size for Two Models. Numbers are overall localization accuracy (%).

91.8% accuracy, motivating our adoption of 50 px as a unified decision boundary. This threshold effectively separates stable, convergent refinement behavior from erroneous drift, providing a simple yet robust criterion for identifying trustworthy iterative localization without additional supervision.

C More Visualization

While the main paper demonstrates the effectiveness of ZoomClick in correcting previously mislocalized predictions through progressive zooming, we further present several misled cases to illustrate

Model	Prezoom Type	D2	D3	D4
Qwen3-VL-32B	No prezoom	67.2	68.9	69.1
	Distance-based	71.3 (+4.1)	72.1 (+3.2)	71.8 (+2.7)
UI-Venus-7B	No prezoom	60.9	63.9	64.1
	Distance-based	62.4 (+1.5)	64.5 (+0.6)	64.8 (+0.7)
UI-Venus-72B	No prezoom	70.5	72.1	72.7
	Distance-based	71.9 (+1.4)	73.1 (+1.0)	72.9 (+0.2)

(a) Effect of Pre-zoom across Zoom Depths (b) Pre-zoom Strategy Comparison

Table 8: Impact of Pre-zoom Strategies. Left: Improvement over no prezoom. Right: Comparison between VLM-based and Distance-based strategies.

Context Strategy	Overall	DEV	CREATIVE	CAD	SCIENTIFIC	OFFICE	OS
None (No Context)	54.1	48.8	50.4	42.1	57.1	77.0	54.1
Prompt Only (Neutral)	49.8	47.8	44.6	37.5	53.5	72.6	46.9
Prompt Only (Positive)	48.3	47.8	44.9	31.4	51.2	71.3	46.9
Visual Marking (Red Cross)	51.0	49.2	45.2	38.7	53.5	73.0	51.0
Full Context Injection	19.0	18.1	20.2	18.0	20.5	22.6	13.3

Table 9: Effect of Context Retention Strategies. Each row corresponds to one of the five designed context mechanisms.

its current limitations. As visualized, two representative failure patterns are observed. First, although a minimum crop size is enforced and the relevant contextual region remains preserved within the view, the model may fail to exploit this context when the cropped distribution deviates from its familiar training distribution, leading to ineffective contextual reasoning and incorrect localization. Second, our method does not explicitly process or restructure the input instruction. For instructions involving sequential or relative semantics (e.g., first, oldest, or spatial comparison among

Model	Shift	Clip	Shrink
Qwen3-VL-32B	72.1	69.6	63.6
UI-Venus-72B	73.0	72.8	70.4
UI-Venus-72B (thres=50)	72.9	73.1	71.2

Table 10: **Effect of different Center-Handling Strategies.**

Pair	Best Threshold (px)	Accuracy	Correct Mean (px)	Error Mean (px)
1-2	33.4	88.2%	13.7	284.9
2-3	47.8	91.1%	8.9	97.8
3-4	52.2	97.0%	3.5	104.9
All (1-4)	50.7	91.8%	8.2	207.0

Table 11: **Optimal distance threshold for correctness classification between consecutive predictions.** We report the best pixel threshold that separates correct and error pairs, along with the resulting accuracy.

adjacent elements*), the model *struggles to correctly interpret the referential intent and is prone to being misled by visually similar neighboring targets*. These cases reveal that beyond visual refinement, improved linguistic understanding and distribution-aware adaptation remain key directions for future enhancement.

Case 1: Context preserved but ineffective due to distribution shift. Figure 5 and 6 represent cases where the minimum crop size successfully retains sufficient surrounding context, and the target region remains fully visible within the view. However, the model still produces incorrect localization. Visually, these cases exhibit atypical visual layouts or uncommon density patterns compared to the model’s training distribution, such as irregular spatial arrangements, cluttered local regions, or uncommon object co-occurrence. Despite the presence of informative context, the model fails to leverage it for disambiguation, resulting in predictions that drift toward visually salient but semantically irrelevant regions. This suggests that context preservation alone is insufficient when the cropped view deviates from the learned distribution manifold.

Case 2: Ambiguous instruction with sequential or comparative semantics. Figure 7 and 8 focus on instruction-driven failures involving sequential or relative semantics, such as “first”, “last”, or references based on positional comparison. In these cases, the visualization shows multiple visually similar elements clustered in close proximity, where the true target differs mainly by ordering or relational position rather than appearance. The model’s prediction tends to gravitate toward the

most visually prominent or centrally located candidate, instead of the semantically correct one indicated by the instruction. This reveals a limitation in understanding instruction-dependent relational cues, causing the system to be misled by nearby distractors with strong visual similarity.

D Pseudocode of Our Method

Algorithm 1: ZoomClick: Patch-Global Pre-zoom with Drift-Free Narrowing

Input: image $I (W \times H)$, instruction q
Output: final click p_{px}
Params: $K=2 \times 2$, threshold τ (px), shrink $\rho \in (0, 1)$, min-crop m , max steps T , mode $\in \{\text{shift}, \text{clip}, \text{shrink}\}$;
 $V \leftarrow (0, 0, 1, 1)$, $I_1 \leftarrow I$, $(W_1, H_1) \leftarrow (W, H)$;
Pre-zoom (patch-global consensus);
 /* Core-1: Selects a cleaner local context when patch and global predictions agree. */
 $p_{dir} \leftarrow \mathcal{G}(I_1, q)$; generate non-overlapping patches $\{I^{(k)}\}_{k=1}^K$;
for $k = 1$ **to** K **do**
 $\hat{p}^{(k)} \leftarrow \mathcal{G}(I^{(k)}, q)$;
 $p^{(k)} \leftarrow \text{MapToOrig}(\hat{p}^{(k)}, \text{patch } k)$;
 $d_k \leftarrow \|p_{dir} - p^{(k)}\|_2$;
 $k^* \leftarrow \arg \min_k d_k$; $p^{(1)} \leftarrow \begin{cases} p^{(k^*)}, & d_{k^*} < \tau \\ p_{dir}, & \text{otherwise} \end{cases}$;
Iterative narrowing (min-crop context retention);
 /* Core-2: Multi-step zoom iteratively narrows the view to refine localization and reach the precise target. */
for $t = 1$ **to** T **do**
 $p_r^{(t)} \leftarrow \text{MapToOrig}(p^{(t)}, V)$;
 if $t=T$ **then**
 return $p_{px} \leftarrow \text{ToPixels}(p_r^{(t)})$
 $(\tilde{W}, \tilde{H}) \leftarrow (\max(\lfloor \rho W_t \rfloor, m), \max(\lfloor \rho H_t \rfloor, m))$;
 $(I_{t+1}, B_t) \leftarrow \text{CropFromOrig}(I, p_r^{(t)}, \tilde{W}, \tilde{H}, \text{mode})$;
 /* Core-3: Min-crop preserves contextual cues; boundary mode controls zoom behavior. */
 $V \leftarrow \text{ComposeViewPort}(V, B_t)$;
 $(W_{t+1}, H_{t+1}) \leftarrow \text{Size}(I_{t+1})$;
 $p^{(t+1)} \leftarrow \mathcal{G}(I_{t+1}, q)$;

E Prompts

Tool-call Prompt for Qwen3-VL Grounding

System: \\
 You are a helpful assistant that can click on elements in UI screenshots. \\
 You are provided with a set of available tools inside XML tags: \\
 \texttt{<tools>\{...\}</tools>}. Each tool is specified by a JSON function signature. \\
 IMPORTANT: You \textbf{must} respond by calling one of these tools to click on the requested element. \\
 \texttt{<tools>\{...\}</tools>}

1086 For each function call, you must return a JSON object wrapped
 1087 in `<tool_call>...</tool_call>` tags: `\\`
 1088 `<tool_call>\\`
 1089 `<name>: <function-name>, "arguments": <args-json-`
 1090 `object>\\`
 1091 `</tool_call>\\`
 1092 Example: To left-click at coordinates (500, 300), respond with:
 1093 `\\`
 1094 `<tool_call>\\`
 1095 `<name>: "computer_use", "arguments": {"action": "`
 1096 `left_click", "coordinate": [500, 300]\\}`
 1097 `</tool_call>\\`
 1098
 1099 User: `\\`
 1100 (1) A UI screenshot image. `\\`
 1101 (2) An instruction: `\\`
 1102 `<name>: "Please click on the element described as: \\`
 1103 `instruction\\}`
 1104 `</tool_call>\\`
 1105 `Respond with a tool call containing the exact pixel`
 1106 `coordinates."`

Prompt for Candidate Selection in Pre-zoom

1107
 1108 You are given a UI screenshot and a user command: `\\`
 1109 `instruction\\}`
 1110 There are `<num_candidates>` candidate UI elements: `\\`
 1111 `candidates\\}`
 1112 Based on the command and the description of candidates, choose
 1113 which candidate better matches the command. `\\`
 1114 Output only the most preferred coordinate in the format `[x1,`
 1115 `x2, y1, y2]`.
 1116

Prompt for UI-Venus-7B Inference

1118
 1119 Outline the position corresponding to the instruction: `\\`
 1120 `instruction\\}`
 1121 The output should be only `[x1, y1, x2, y2]`.
 1122

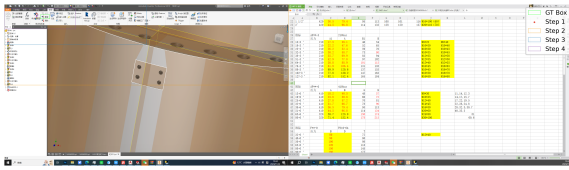


Figure 5: "Update style."

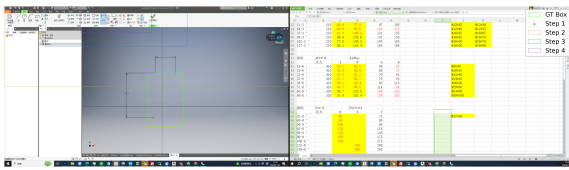


Figure 6: "Overlap Constraint."

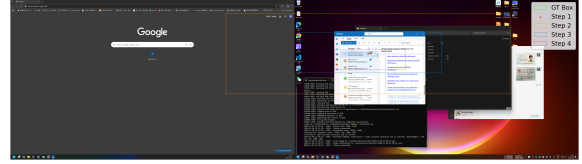


Figure 7: "Unpin the first mail entry."

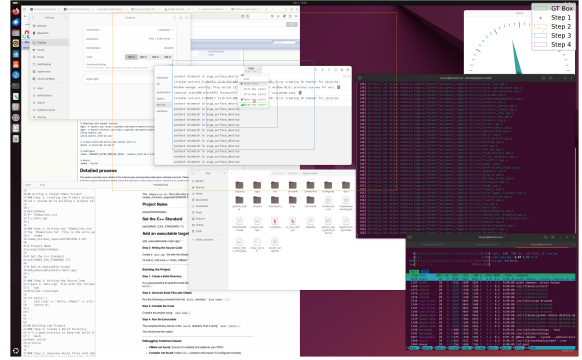


Figure 8: "View oldest logs."

F Qualitative Results

1124



Figure 9: **Qualitative results of ZOOMCLICK on ScreenSpot-Pro.** Left: depth-2 zoom resolves the error from depth 1. Center: depth-3 zoom resolves the errors from depths 1-2. Right: depth-4 zoom resolves the errors from depths 1-3.

G Potential Risks

1125

1126 Our GUI-level interactions may expose sensitive
 1127 on-screen information and introduce privacy risks.
 1128 In addition, our approach relies on large models
 1129 whose computation contributes to non-negligible
 1130 energy consumption.