AUTOFOCUS: UNCERTAINTY-AWARE ACTIVE VISUAL SEARCH FOR GUI GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

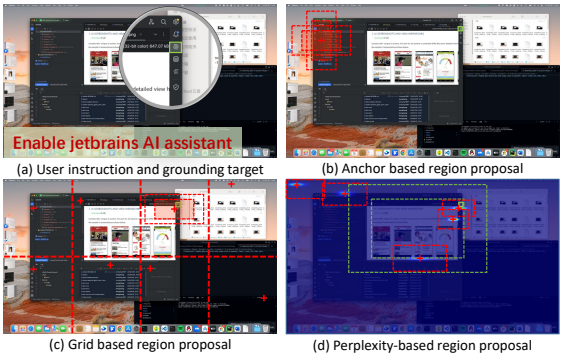
Vision-Language Models (VLMs) have enabled autonomous GUI agents that translate natural language instructions into executable screen coordinates. However, grounding performance degrades in high-resolution interfaces, where dense layouts and small interactive elements expose a resolution gap between modern displays and model input constraints. Recent methods have explored zoom-in refinement, confidence-based selection, and reward-driven optimization. But most of them either rely on manually designed crop policies, consensus over sampled predictions, or additional reinforcement learning objectives. In contrast, we study whether the model's own autoregressive coordinate distribution can serve as an intrinsic signal for active visual search at test time. We propose AutoFocus, a training-free uncertainty-aware refinement framework that converts token-level coordinate perplexity into an anisotropic spatial probability field. This field guides global-local region proposal generation, while an error-triggered verifier selectively activates refinement only when the initial prediction is uncertain. AutoFocus therefore improves high-resolution GUI grounding without requiring additional annotations, reward modeling, or parameter updates. Extensive experiments on ScreenSpot-Pro, ScreenSpot-V2 and OSWorld-G demonstrate consistent improvements across both general-purpose and GUI-specialized VLMs.

1 INTRODUCTION

The synergy between Vision Language Models (VLMs) and Graphical User Interface (GUI) automation has paved the way for autonomous agents capable of executing complex tasks across digital platforms. These agents Yuan et al. (2025a); Huang et al. (2025); Sun et al. (2025) aim to bridge the "intention-action gap" by transforming natural language instructions into on-screen operations, such as identifying a specific pixel coordinate to execute an action based on the textual instruction.

However, despite the remarkable reasoning capabilities of frontier VLMs, they still lack precise grounding ability when deployed in professional, high-resolution environments. This failure stems from a fundamental resolution mismatch. Modern interfaces operate at $1080p$ or $4K$ resolutions, where critical interactive elements may occupy only a few dozen pixels. In contrast, VLMs typically process resized inputs with constrained visual tokens, limiting their ability to resolve fine-grained details. As a result, a model may semantically identify the correct UI component yet fail to localize it precisely, producing coordinate offsets or missing small icons altogether.

Figure 1: Comparison with other zoom-in methods.



Recent works have explored multi-stage "zoom-in" strategies to mitigate this problem. However, existing approaches often rely on fixed anchors, heuristic grids, or zooming around a single initial prediction (as shown in Figure 1), limiting adaptability across varying GUI scales Luo et al. (2025);

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.

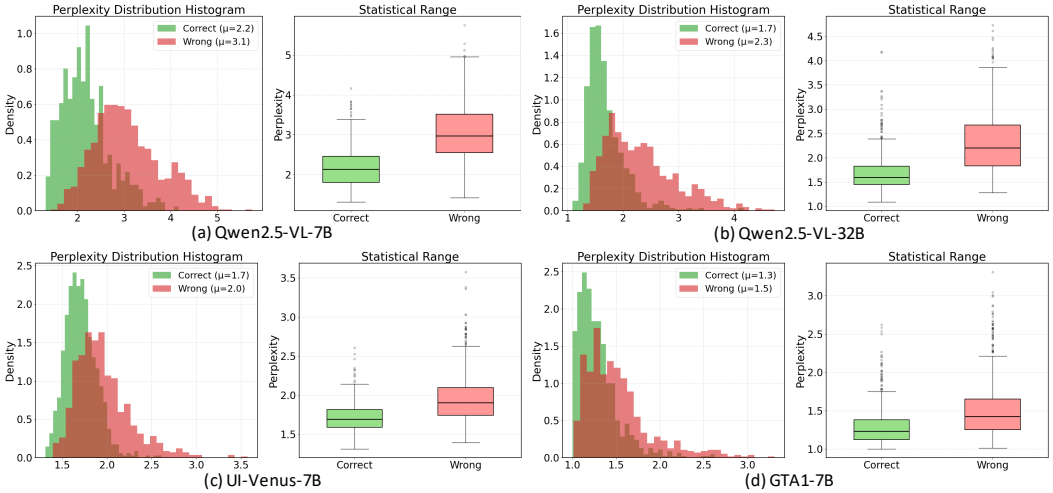


Figure 2: Perplexity (PPL) distribution of grounding results on ScreenSpot-Pro. We compare the PPL across multiple models, with green and red areas indicating correct and incorrect predictions, respectively. Higher PPL values for incorrect results suggest a strong correlation between model uncertainty and grounding failure.

Wu et al. (2025a); Park et al. (2025); Jiang et al. (2025). Other methods employ reinforcement learning to optimize cropping policies Lei et al. (2025); Ye et al. (2025), but at the cost of significant training overhead and environmental interaction. These solutions lack a unified and model-intrinsic criterion for adaptive refinement.

In this work, our key insight is that token-level perplexity (PPL) in coordinate generation correlates with spatial ambiguity. Rather than treating perplexity purely as a language modeling metric, we reinterpret it as an intrinsic uncertainty signal reflecting the model's confidence in spatial localization. We empirically validate this assumption in Figure 2, where we analyze the relationship between token-level perplexity and grounding correctness across multiple backbone models. As shown, incorrect predictions consistently exhibit significantly higher axial perplexity than correct ones, with clear distributional separation in both histogram and boxplot statistics. This trend holds across general-purpose (Qwen2.5-VL Bai et al. (2025)) and GUI-specialized models (UI-Venus Gu et al. (2025), GTA1 Yang et al. (2025a)), indicating that perplexity serves as a reliable intrinsic proxy for spatial uncertainty.

Based on this, we propose AutoFocus, a training-free, uncertainty-aware active visual search framework for high-resolution GUI grounding. AutoFocus first determines whether refinement is necessary based on the model's initial prediction and its associated uncertainty. If the prediction exhibits high ambiguity, the core Gaussian Dynamic Focusing (GDF) module is activated. Specifically, we map the x- and y-axis perplexities into anisotropic variances (σ_x, σ_y), forming a continuous 2D Gaussian probability field over the high-resolution screenshot. This density field models directional uncertainty, enabling the agent to reason about both where ambiguity lies and how uncertainty spreads spatially. Based on this field, AutoFocus generates global and local region proposals and introduces Shape-Aware Zooming, which adaptively balances tight localization with contextual preservation. Finally, predictions are performed on these region proposals and a structured aggregation stage selects the most consistent prediction.

By leveraging intrinsic uncertainty signals already available within the model, AutoFocus dynamically tailors its "foveal" search area and provides a principled and computationally controllable refinement mechanism for high-resolution grounding. While a contemporaneous work, UI-Zoomer Tang et al. (2026b), also explores uncertainty-driven adaptive zoom-in, it derives uncertainty from inter-sample spatial variance and scalar token confidence to produce a single isotropic square crop. In contrast, AutoFocus leverages axial token-level perplexity to construct an anisotropic Gaussian probability field, enabling direction-aware region proposals that better match GUI element geometry, and further employs multi-proposal visual aggregation for more robust refinement. This design also makes AutoFocus easy to integrate into existing VLM inference pipelines, since it only relies on token likelihoods that are commonly exposed during autoregressive decoding. We conduct

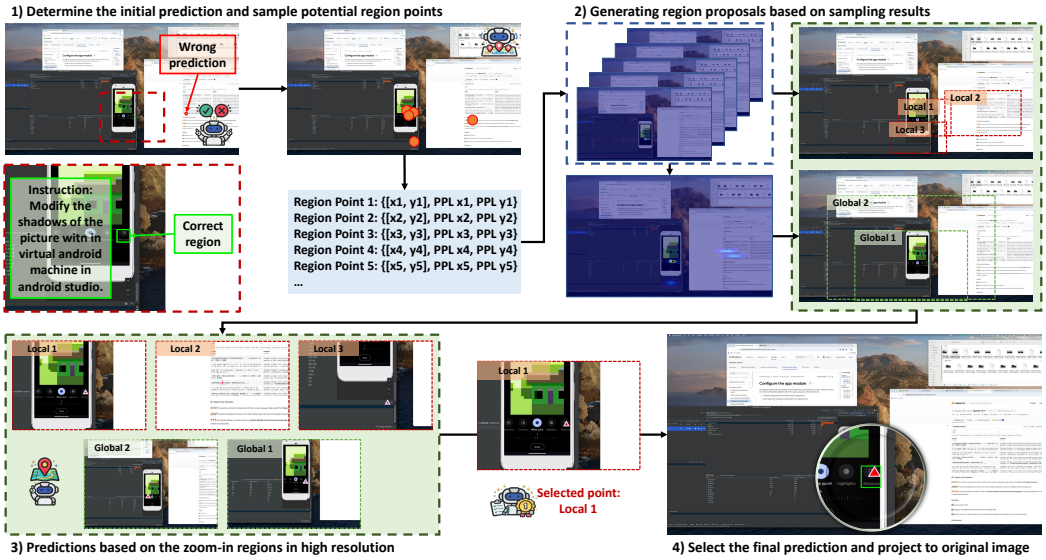


Figure 3: Overview of AutoFocus. The model first samples multiple coordinate hypotheses with axial perplexities from the initial prediction. Gaussian Dynamic Focusing constructs an anisotropic spatial field to generate global and local proposals. High-resolution zoom-in predictions are performed on candidate regions, and a structured aggregation step selects the final coordinate.

comprehensive evaluations on the ScreenSpot-V2, ScreenSpot-Pro and OSWorld-G benchmarks. AutoFocus consistently delivers substantial performance gains across various base models.

2 Related Work

2.1 GUI Agents

Recent advancements in Vision Language Models (VLMs) have significantly enhanced GUI automation, enabling agents to navigate graphical user interfaces with increasing sophistication Deng et al. (2023); Gou et al. (2024); Pahuja et al. (2025); Wang et al. (2024); Chen et al. (2025b;a). Prior studies generally adopt text-based reasoning, using structured representations such as HTML or accessibility trees Koh et al. (2024); Cao et al. (2024). Some methods also use LLMs to extract structured interface information Yu et al. (2025) and supplementary textual details for agents input Zheng et al. (2024). As the field moves toward a more "human-like" embodiment and the enhancement of the VLMs' visual reasoning capabilities, recent methods emphasize direct visual interpretation of GUI elements from raw pixels Gou et al. (2024); Qin et al. (2025). Despite these strides, they often struggle with high-resolution or ambiguous professional interfaces Li et al. (2025b); Wu et al. (2024), where diminutive icons and dense layouts degrade grounding reliability. More recent studies have begun to exploit uncertainty, confidence, or reward signals for GUI grounding. For example, GUI-RCPO Du et al. (2026) estimates implicit confidence from the agreement among multiple sampled predictions. GUI-G^2 Tang et al. (2026b) formulate GUI grounding as a reward modeling problem and use Gaussian distributions centered on target elements to provide dense training signals. However, the prevailing reliance on supervised fine-tuning and reinforcement learning poses limitations in scalability and adaptability, motivating a paradigm shift toward test-time scaling, aims to overcome the need for vast labeled datasets. **Please see the Appendix A.1 for more related work.**

3 Method

In this section, we present AutoFocus, a training-free, uncertainty-aware active visual search framework for GUI grounding, the overall framework is depicted in Figure 3.

3.1 ERROR-TRIGGERED ACTIVE REFINEMENT

Given a text description and the original image, AutoFocus first performs an initial coarse grounding prediction. Instead of blindly applying zoom-in refinement to every prediction, AutoFocus employs a visual prompt-based self-verification mechanism to determine whether refinement is necessary. This allows the framework to selectively allocate computational resources to uncertain cases while avoiding redundant zoom-in operations for confident predictions. Specifically, given the initial prediction $p_0 = (x_0, y_0)$, we construct a visual-prompted image $I_v = \mathcal{V}(I_{raw}, p_0)$, where $\mathcal{V}$ denotes a drawing operation that overlays a salient visual marker (e.g., a colored point or star Luo et al. (2025)) at the predicted location. The prompted image is then fed back into the VLM together with a verification query, such as: *"Does the marked point correctly correspond to the target element described in the instruction?"*

Based on the model's response, AutoFocus determines whether the prediction is sufficiently reliable. If the model confirms the prediction, the coordinate is directly accepted as the final result. Otherwise, the prediction is considered uncertain, and the Gaussian Dynamic Focusing (GDF) module is activated to construct a spatial uncertainty field and perform targeted zoom-in refinement. This visual prompt-based verification converts the refinement trigger into a structured visual reasoning task, enabling AutoFocus to dynamically and efficiently focus on ambiguous regions.

3.2 GAUSSIAN DYNAMIC FOCUSING

When visual self-verification determines that the initial prediction is unreliable, AutoFocus activates the Gaussian Dynamic Focusing (GDF) module to perform uncertainty-guided spatial exploration. Unlike conventional grounding methods that rely on a single deterministic prediction, GDF explicitly models the spatial uncertainty inherent in the VLM's generative process and converts it into a structured spatial probability field. This enables principled region proposal generation that reflects both prediction confidence and directional ambiguity. GDF consists of three stages: (1) Probabilistic coordinate sampling with axial perplexity estimation, (2) Anisotropic gaussian splatting to construct a continuous spatial density field, and (3) Global-local region proposal generation.

Probabilistic coordinate sampling and axial perplexity. Modern VLMs formulate coordinate grounding as an autoregressive token generation process over a discrete spatial vocabulary. Given an image $\mathcal{I}$ and instruction $\mathcal{T}$, the model produces a sequence of coordinate tokens whose likelihood reflects its confidence in the prediction. While greedy decoding yields only a single point estimate, selecting the token with the highest posteriori probability as the final prediction, it discards valuable information about prediction uncertainty. To recover this information, we perform stochastic decoding with temperature τ to obtain a set of N sampled coordinates:

$$\mathcal{S} = \{p_i\}_{i=1}^{N}, \quad p_i = (x_i, y_i) \tag{1}$$

where each p_i represents the i-th sampled coordinate of potential target locations based on the instruction. Each coordinate is generated as a token sequence, which we decompose into independent axial subsequences for the horizontal and vertical dimensions:

$$\mathbf{t}_x = \{t_{x,1}, \ldots, t_{x,L_x}\}, \quad \mathbf{t}_y = \{t_{y,1}, \ldots, t_{y,L_y}\} \tag{2}$$

where L_x and L_y denote the length of two axial subsequences $\mathbf{t}_x$ and $\mathbf{t}_y$. where L_x and L_y denote the lengths of the axial subsequences $\mathbf{t}_x$ and $\mathbf{t}_y$, i.e., the numbers of tokens used to encode the x and y coordinates, respectively. Then, we compute axial perplexity as a measure of uncertainty along each spatial dimension:

$$PPL_x^{(i)} = \exp\left(-\frac{1}{L_x}\sum_{j=1}^{L_x}\log P(t_{x,j} \mid \mathbf{t}_{x,<j}, \mathcal{I}, \mathcal{T})\right), \tag{3}$$

$$PPL_y^{(i)} = \exp\left(-\frac{1}{L_y}\sum_{j=1}^{L_y}\log P(t_{y,j} \mid \mathbf{t}_{y,<j}, \mathbf{t}_x, \mathcal{I}, \mathcal{T})\right). \tag{4}$$

where $PPL_x^{(i)}$ and $PPL_y^{(i)}$ denotes the perplexity of the i-th sampled coordinate on axis x and y. Perplexity serves as an intrinsic uncertainty signal derived directly from the model's generative

process. Higher perplexity indicates lower confidence and suggests that the true target location may lie within a broader spatial neighborhood.

Importantly, axial decomposition enables directional uncertainty modeling. GUI elements such as text fields and sliders often exhibit anisotropic geometry, where uncertainty differs across axes. Modeling these independently allows more precise spatial reasoning.

Anisotropic gaussian splatting. To bridge the gap between discrete coordinate samples and the underlying continuous spatial distribution, we project the sampled hypotheses into a continuous probability field via Anisotropic Gaussian Splatting. Each sampled coordinate is modeled as an anisotropic Gaussian kernel:

$$\mathcal{N}_i(p) = \exp\left(-\frac{1}{2}(p-\mu_i)^T\Sigma_i^{-1}(p-\mu_i)\right), \tag{5}$$

where $\mu_i = (x_i, y_i)$ represents the sampled centroid. The covariance matrix Σ_i is decoupled to explicitly reflect the directional uncertainty of the model.

$$\mu_i = (x_i, y_i), \quad \Sigma_i = \begin{bmatrix} \sigma_{x,i}^2 & 0 \\ 0 & \sigma_{y,i}^2 \end{bmatrix}, \tag{6}$$

and the covariance is derived from axial perplexity:

$$\sigma_{z,i} = \beta \cdot PPL_z^{(i)}, \quad z \in \{x, y\}. \tag{7}$$

Here, the hyperparameter β modulates the exploration scale, allowing the kernel to expand along axes where the model exhibits higher perplexity. This anisotropic formulation is critical for capturing the structural priors of GUI elements, such as elongated text fields or narrow navigation bars. Finally, the individual kernels are synthesized into a global spatial density field $M(p)$ through a weighted aggregation:

$$M(p) = \sum_{i=1}^{N} w_i, \mathcal{N}_i(p). \tag{8}$$

The influence of each sample is governed by its relative confidence w_i, computed as the softmax-normalized negative sentence-level perplexity of the i-th prediction:

$$w_i = \frac{\exp(-PPL_i)}{\sum_j^N \exp(-PPL_j)}. \tag{9}$$

By prioritizing semantically stable samples while accounting for spatial uncertainty, the resulting density field $M(p)$ provides a robust foundation for identifying the most salient candidate regions for zoom-in refinement.

Global-local region proposal generation. Based on the density field, we generate region proposals at two complementary scales.

Local proposals. Each Gaussian kernel defines a spatial hypothesis B_i representing the 3σ confidence interval of the prediction:

$$B_i = [x_i - 3\sigma_{x,i},\ x_i + 3\sigma_{x,i}] \times [y_i - 3\sigma_{y,i},\ y_i + 3\sigma_{y,i}]. \tag{10}$$

We then apply Non-Maximum Suppression (NMS) to prune redundant proposals, yielding the K_{local} most salient regions. The confidence score for NMS is derived from the model's total response perplexity, ensuring the refinement stage focuses on the most semantically stable predictions.

Global proposals. The global region proposal is derived from the aggregated spatial density field $M(p)$, which encodes the overall uncertainty-aware distribution over the image plane. We approximate this density field using moment matching and compute the mean and covariance as:

$$\mu = \frac{\sum_p M(p)\, p}{\sum_p M(p)}, \Sigma = \frac{\sum_p M(p)\,(p-\mu)(p-\mu)^T}{\sum_p M(p)} \tag{11}$$

The center of the global proposal is defined as the mean μ, while its width and height are determined by the standard deviations along each axis:

$$W = \alpha\sigma_x, \quad H = \alpha\sigma_y \tag{12}$$

where $\sigma_x = \sqrt{\Sigma_{xx}}$ and $\sigma_y = \sqrt{\Sigma_{yy}}$, and α is a scale factor controlling the coverage of the global proposal region. When multiple global proposals are required, different scale factors can be used to construct a hierarchy of K_{global} regions capturing varying levels of spatial context.

Since the resulting proposal may extend beyond image boundaries, we apply a boundary-aware adjustment that translates the proposal toward the image center while preserving its dimensions. This ensures that the proposal remains fully contained within the valid image region without altering its uncertainty-derived spatial extent. To further balance precision and contextual coverage, we introduce Shape-Aware Zooming (SAZ), which interpolates between anisotropic and square crops:

$$W_{final} = W_{raw} + \lambda(\max(W_{raw}, H_{raw}) - W_{raw}), \tag{13}$$

$$H_{final} = H_{raw} + \lambda(\max(W_{raw}, H_{raw}) - H_{raw}), \tag{14}$$

where $\lambda \in [0, 1]$ controls the trade-off between tight localization and contextual completeness.

3.3 Action Prediction and Aggregation

Given the set of global and local region proposals generated by GDF, AutoFocus performs a second-pass high-resolution grounding step followed by a visual aggregation stage to produce the final action coordinate. For each proposal region R_j, where $j \in \{1, \ldots, K\}$ and $K = K_{\text{global}} + K_{\text{local}}$, we independently apply the VLM to perform refined grounding under a zoomed-in view. Specifically, each region R_j is cropped from the original image and resized to the model's standard input resolution, enabling the VLM to access fine-grained visual details that may be indistinguishable at the original scale. The resized crop, together with the original instruction $\mathcal{T}$, is fed into the VLM to produce a refined coordinate prediction: $\hat{c}_j = (\hat{x}_j, \hat{y}_j)$, where $(\hat{x}_j, \hat{y}_j)$ denotes the predicted coordinate in the local coordinate system of the resized crop.

Each refined prediction is then projected back into the original image coordinate system. Let $(x_j^{\min}, y_j^{\min})$ denote the top-left corner of region R_j in the original image, and let $(s_{x,j}, s_{y,j})$ denote the scaling factors between the original crop and the resized input along the horizontal and vertical axes, respectively. The global coordinate is computed as:

$$c_j = \left(x_j^{min} + \frac{\hat{x}_j}{s_{x,j}},\ y_j^{min} + \frac{\hat{y}_j}{s_{y,j}} \right). \tag{15}$$

This yields a set of candidate coordinates $\{c_j\}_{j=1}^{K}$, each corresponding to a refined hypothesis derived from a distinct uncertainty-guided region.

Instead of selecting the final coordinate based solely on numerical confidence, AutoFocus formulates the aggregation step as a multi-image visual reasoning task. For each candidate coordinate c_j, we construct an annotated image $\tilde{\mathcal{I}}_j = \mathcal{V}(\mathcal{I}, c_j)$ by overlaying a visual marker at the corresponding location on the original image. This results in a set of annotated images:

$$\tilde{\mathcal{I}} = \{\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_K\}. \tag{16}$$

These annotated images are jointly provided to the VLM together with the original instruction $\mathcal{T}$, leveraging the model's native multi-image reasoning capability. The model is prompted to select the image whose marked location best corresponds to the target element. Formally, the VLM produces a categorical distribution over the candidate images:

$$j^* = \arg\max_j P(j \mid \tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_K, \mathcal{T}), \tag{17}$$

and the final predicted coordinate is defined as $c^* = c_{j^*}$, where j^* corresponds to the candidate whose marked position best aligns with the instruction. This aggregation strategy transforms the final grounding step into a structured comparison task, allowing the model to leverage contextual reasoning across multiple hypotheses.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Base Models. In the following experiments, we integrate our proposed AutoFocus pipeline with four representative Vision-Language Models (VLMs): (i) UI-TARS Qin et al. (2025), a model specifically trained for GUI grounding and interaction; (ii) Qwen2.5-VL Bai et al. (2025), a strong general-purpose multimodal foundation model; and (iii) GTA1-Qwen Yang et al. (2025a), UI-Venus Gu et al. (2025) and UI-Venus-2 Team et al. (2026) three GUI-specialized agent built upon Qwen2.5-VL and Qwen3.5 with reinforcement learning tailored for UI interaction tasks.

Hyperparameter. Please refer to the Appendix A.2 for hyperparameter settings.

Benchmarks. We evaluate our method on GUI grounding benchmarks covering both OS-level operation tasks and Web interaction scenarios. Specifically, we adopt ScreenSpot-Pro Li et al. (2025b), ScreenSpot-V2 Wu et al. (2024), and OSWorld-G Xie et al. (2025), and report performance using the standard *Accuracy* metric.

4.2 MAIN RESULTS

Table 1: Comparison of various models on ScreenSpot-Pro Li et al. (2025b). The final average scores are in **bold**. Results of the three base models are highlighted in blue, and their corresponding AutoFocus-enhanced variants are highlighted in red.

Method	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
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**
Seed 2.1 Pro Seed (2026)	-	-	-	-	-	-	-	-	-	-	-	61.4	-	-	-	-	-	-	-	-	**65.3**
Aria-UI Yang et al. (2025b)	16.2	0.0	8.4	23.7	2.1	14.7	7.6	1.6	6.1	27.1	6.4	18.1	20.3	1.9	16.1	4.7	0.0	2.6	17.1	2.0	**11.3**
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**
InfiGUI-R1-3B Liu et al. (2025a)	51.3	12.4	32.4	44.9	7.0	29.0	33.0	14.1	28.4	58.3	20.0	41.7	65.5	28.3	57.0	43.9	12.4	29.6	49.1	14.1	**35.7**
InfiGUI-G1-3B Liu et al. (2025b)	64.9	20.0	-	51.5	16.8	-	50.8	25.0	-	68.8	32.7	-	70.6	32.1	-	49.5	15.7	-	-	-	**45.2**
SE-GUI-3B Yuan et al. (2025b)	55.8	7.6	35.1	47.0	4.9	29.0	38.1	12.5	31.8	61.8	16.4	43.3	59.9	24.5	50.9	40.2	12.4	25.5	50.4	11.8	**35.9**
Jedi-3B Xie et al. (2025)	61.0	13.8	38.1	53.5	8.4	34.6	27.4	9.4	23.0	54.2	18.2	38.6	64.4	32.1	57.0	38.3	9.0	25.0	49.8	13.7	**36.1**
GUI-G1-3B Zhou et al. (2025)	50.7	10.3	31.1	36.6	11.9	26.6	39.6	9.4	32.2	61.8	30.0	48.0	67.2	32.1	59.1	23.5	10.6	16.1	49.5	16.8	**37.1**
Dimo-GUI-7B Wu et al. (2025a)	66.9	21.4	44.8	60.6	21.7	44.3	50.3	14.1	41.4	68.1	21.8	48.0	80.8	52.8	74.3	69.2	28.1	50.5	65.2	24.5	49.7
RegionFocus-7B Luo et al. (2025)	53.2	3.4	29.1	42.9	4.9	27.0	28.4	3.1	22.2	56.9	10.9	37.0	59.9	24.5	51.7	41.1	15.7	29.6	46.6	8.8	**32.1**
RegionFocus-72B Luo et al. (2025)	75.3	25.5	51.2	76.3	30.8	57.2	71.6	28.1	60.9	87.5	39.1	66.5	87.0	60.4	80.9	74.8	36.0	57.1	78.6	34.1	**61.6**
UI-Zoomer Tang et al. (2026a)	80.5	37.2	-	70.1	31.5	-	77.2	34.4	-	82.6	30.0	-	88.8	50.9	-	67.3	37.1	-	78.1	35.4	**61.8**
UI-TARS-7B Qin et al. (2025)	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**
+ *AutoFocus*	64.9	13.1	39.4	63.6	14.7	43.1	43.7	14.1	37.2	78.7	30.3	57.6	75.7	35.8	66.5	57.7	20.8	40.3	63.2	15.8	**45.1**
UI-TARS-72B Qin et al. (2025)	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**
+ *AutoFocus*	74.5	28.7	51.9	65.1	23.9	47.1	45.7	14.1	38.5	78.7	30.3	57.6	83.4	37.8	70.5	60.7	19.1	41.8	66.0	30.9	**52.6**
Qwen2.5-VL-7B Bai et al. (2025)	45.5	1.4	24.1	32.8	6.3	21.7	22.3	6.2	18.4	50.7	7.3	31.9	52.5	15.1	43.9	36.4	10.1	24.5	39.3	6.6	**26.8**
+ *AutoFocus*	57.5	6.4	33.1	46.8	10.0	31.1	36.2	7.9	29.3	58.3	11.0	37.9	62.3	23.5	53.5	40.2	16.9	29.6	50.2	11.3	**35.4**
Qwen2.5-VL-72B Bai et al. (2025)	66.2	13.8	40.8	64.6	15.4	44.0	47.7	12.5	39.1	78.5	29.1	57.1	74.6	37.7	66.1	60.7	22.5	43.4	64.9	20.2	**47.8**
+ *AutoFocus*	81.2	38.1	61.1	77.3	36.5	60.6	73.1	29.0	62.4	91.0	39.3	68.9	88.6	62.7	82.8	75.2	34.5	56.8	80.9	39.5	**65.1**
GTA1-Qwen-7B Yang et al. (2025a)	66.9	20.7	44.5	62.6	18.2	44.0	53.3	17.2	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**
+ *AutoFocus*	70.5	23.8	47.3	65.2	18.2	45.5	58.9	26.6	51.0	78.5	30.0	57.5	84.1	52.6	76.4	55.6	30.7	44.3	69.5	31.0	**54.8**
GTA1-Qwen-32B Yang et al. (2025a)	82.5	28.3	56.2	69.2	14.7	46.3	43.7	23.4	38.7	79.9	31.8	59.1	80.8	43.4	72.2	70.1	32.6	53.1	69.9	27.2	**53.6**
+ *AutoFocus*	87.0	33.8	61.2	73.2	25.2	53.1	59.9	31.2	52.9	84.7	34.5	63.0	85.9	49.1	77.4	75.7	51.4	65.2	77.0	40.4	**63.0**
UI-Venus-7B Gu et al. (2025)	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**
+ *AutoFocus*	84.7	35.7	63.6	75.5	37.0	60.2	82.2	43.8	73.0	90.5	51.2	75.8	88.6	59.2	83.3	78.1	39.7	61.5	82.1	44.7	**67.8**
UI-Venus2-9B Team et al. (2026)	85.7	60.4	73.5	76.1	50.0	65.2	80.7	48.4	72.8	84.0	47.3	68.1	85.8	75.0	83.3	81.3	57.3	70.4	82.1	56.2	**72.2**
+ *AutoFocus*	84.4	65.3	75.2	76.6	61.3	70.2	84.3	48.4	75.5	88.9	55.5	74.4	89.2	82.7	87.7	82.2	55.1	71.5	84.1	60.7	**75.6**

4.2.1 SCREENSPOT-PRO (SS-PRO)

Table 1 reports results on the challenging ScreenSpot-Pro benchmark, which features high-resolution professional software interfaces with dense layouts and small interactive elements. AutoFocus consistently improves all base models across every domain category (Development, Creative, CAD, Scientific, Office, and OS). As detailed in the results, AutoFocus enhances UI-TARS-7B by 9.4% absolute, a scaling effect that becomes even more pronounced in the 72B variant. Similar gains are observed for Qwen2.5-VL family: the 7B variant rises from 26.8% to 35.4%, while the 72B model improves substantially from 47.8% to 65.1% (+17.3). Notably, Qwen2.5-VL-72B + AutoFocus achieves an average accuracy of 65.1%, surpassing strong zoom-based baselines such as RegionFocus-72B (61.6%). The framework also proves highly effective for RL-tuned agents, pushing GTA1-Qwen-32B to 63.0% accuracy. Furthermore, the GUI-specialized UI-Venus-7B experiences a dramatic performance leap, with its average accuracy surging from 50.3% to 67.8% (+17.5), demonstrating that even models already optimized for UI grounding benefit profoundly

Table 2: Comparison with state-of-the-art methods on ScreenSpot-V2 Wu et al. (2024) across mobile, desktop, and web domains. We report grounding accuracy (%) categorized by grounding target type: Text, Icon/Widget, and the overall Average (Avg). The final average scores are in **bold**. The base model results are highlighted in blue, and their corresponding AutoFocus-enhanced variants are highlighted in red.

Agent Model	Mobile		Desktop		Web		Avg
	Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
OS-Atlas-Base-7B Wu et al. (2024)	96.2	83.4	89.7	69.3	94.0	79.8	**87.1**
Jedi-3B Xie et al. (2025)	96.6	81.5	96.9	78.6	88.5	83.7	**88.6**
UI-TARS-1.5-7B Qin et al. (2025)	95.9	84.8	94.9	80.7	90.6	86.2	**89.7**
UI-TARS-72B Qin et al. (2025)	94.8	86.3	91.2	87.9	91.5	87.7	**90.3**
Jedi-7B Xie et al. (2025)	96.9	87.2	95.9	87.9	94.4	84.2	**91.7**
GUI-Actor-7B Wu et al. (2025b)	97.6	88.2	96.9	85.7	93.2	86.7	**92.1**
Qwen2.5-VL-7B Bai et al. (2025)	97.6	87.2	90.2	74.2	93.2	81.3	**88.8**
+ AutoFocus	98.8	88.9	95.6	85.5	96.2	91.2	**93.7**
Qwen2.5-VL-72B Bai et al. (2025)	99.0	90.1	96.4	87.1	96.6	90.6	**94.0**
+ AutoFocus	99.0	93.2	97.5	89.3	96.4	92.7	**95.5**
UI-Venus-7B Gu et al. (2025)	99.0	90.0	97.0	90.7	96.2	88.7	**94.1**
+ AutoFocus	99.3	91.4	97.5	90.7	96.8	91.2	**95.3**

Table 3: Performance comparison Qin et al. (2025); Gou et al. (2025); Xu et al. (2024); Wu et al. (2025b); Ouyang et al. (2026) on OSWorld-G Xie et al. (2025).

Model	Text	Elem	Layout	Manip	Refuse	Avg.
UI-TARS-7B	60.2	51.8	54.9	35.6	0.0	47.5
UGround-7B	51.3	40.3	43.5	24.8	0.0	36.4
Aguvis-7B	55.9	41.2	43.9	28.2	0.0	38.7
GUI-Actor-7B	60.2	54.2	58.1	30.9	0.0	49.5
Jedi-3B	67.4	53.0	53.8	44.3	7.4	50.9
Jedi-7B	65.9	55.5	57.7	46.9	7.4	54.1
FocusUI-7B	63.6	61.2	63.6	34.9	0.0	54.4
Qwen2.5-VL-7B	45.6	32.7	41.9	18.1	0.0	31.4
+ AutoFocus	52.5	40.3	44.5	23.6	0.0	37.3
Qwen2.5-VL-32B	63.2	47.3	49.0	36.9	0.0	46.5
+ AutoFocus	69.3	58.7	58.6	49.1	0.0	56.3

from our uncertainty-guided visual refinement strategy. It also surpass the contemporaneous UI-Zoomer (61.8%) by 6.0% on the same backbone. The improvement is particularly pronounced on text targets, where AutoFocus's anisotropic spatial modeling better captures the directional uncertainty of elongated text elements, while UI-Zoomer's isotropic square crops may include excessive background or miss horizontal context.

4.2.2 SCREENSPOT-V2 (SS-V2)

Table 2 presents results on ScreenSpot-V2 across mobile, desktop, and web environments. Qwen2.5-VL-7B + AutoFocus achieves an overall accuracy of 93.7%, competitive with strong 72B baselines. Qwen2.5-VL-72B + AutoFocus further reaches 95.5%, outperforming the baseline Qwen2.5-VL-72B (94.0%). Notably, even with a base model that already achieves high-saturation accuracy (UI-Venus), our framework provides a substantial boost, raising the overall average from 94.1% to 95.3%. Gains are consistent across both Text and Icon categories, with particularly strong improvements in desktop and web domain. Together, these results demonstrate that AutoFocus delivers robust and scalable improvements across both high-resolution professional interfaces and general GUI environments, validating the effectiveness of uncertainty-guided active visual refinement.

4.2.3 OSWORLD-G

As shown in Table 3, AutoFocus consistently improves Qwen2.5-VL backbones on OSWorld-G. For Qwen2.5-VL-7B, AutoFocus increases the average score from 31.4 to 37.3, with gains across various categories. The improvement becomes more pronounced on Qwen2.5-VL-32B, where AutoFocus raises the average score from 46.5 to 56.3.

4.3 Ablation Studies

To rigorously understand which design choices contribute to AutoFocus, we conduct ablations along two axes: (i) **component-level** ablations to validate the necessity of each module in the pipeline, and (ii) **hyperparameter** ablations (Please refer to the Appendix A.3) to quantify sensitivity to inference-time settings, e.g., sampling budget and zoom scale. Unless otherwise specified, we report *Accuracy* on ScreenSpot-Pro and ScreenSpot-V2 to ensure fair compute comparisons.

4.3.1 Component ablations (What matters?)

To dissect the contribution of each module within the AutoFocus framework, we evaluate a series of ablation variants by progressively integrating or isolating its key components. This systematic analysis, summarized in Table 4, reveals a clear evolutionary trajectory from deterministic point prediction to uncertainty-aware region refinement.

We first establish a **Baseline (no zoom)** using standard single-pass coordinate prediction, which achieves a modest 26.8% accuracy on ScreenSpot-Pro, severely constrained by the resolution bottleneck. Simply introducing **+ Multi-sample only**, where we sample N candidates but execute the most confident one without any spatial modeling, yields marginal gains (+0.7%), suggesting that stochastic sampling alone cannot resolve grounding ambiguities. A significant performance leap occurs with the introduction of the probability density field and zoom-in strategy. Transitioning to the **+ Global proposals** provides a +4.1% boost, while **+ Local proposals** improves accuracy to 33.1%. These results demonstrate that shifting the model's focus from discrete points to structured regional hypotheses is fundamental to overcoming high-resolution visual clutter. We further ablate the sophisticated design choices that distinguish AutoFocus from naive zooming mechanisms:

Table 4: Ablations with Qwen2.5-VL-7B.

Variant	SS-Pro ↑	SS-V2 ↑
Baseline (no zoom)	26.8	88.8
+ Multi-sample only	27.5	89.4
+ Global proposals	31.6	90.9
+ Local proposals	33.1	92.3
w/o Axial PPL	34.1	92.9
w/o SAZ	33.8	92.4
w/o Vis-Aggregation	31.6	91.1
AutoFocus (Full)	**35.4**	**93.7**

Importance of Structural Priors (w/o Axial PPL): Replacing directional uncertainty (PPL_x, PPL_y) with a global sequence perplexity leads to a performance drop (34.1% vs. 35.4%). This validates that GUI elements possess strong structural anisotropy; modeling x and y uncertainty independently is crucial for capturing elongated elements such as text bars or vertical menus.

Precision vs. Context (w/o SAZ): Disabling Shape-Aware Zooming forces the model to use raw, uncertainty-derived crops. The resulting degradation (33.8%) highlights the necessity of SAZ in balancing character-level precision with the semantic context required for high-level reasoning.

Rational Decision Making (w/o Visual Aggregation): The most dramatic drop occurs when replacing our multi-region visual aggregation with naive confidence-based selection, with accuracy falling back to 31.6%. This suggests that the "second look" provided by AutoFocus is effective when regional insights are synthesized rather than merely filtered.

4.3.2 Hyperparameter ablations.

Please refer to the Appendix A.3 for more details.

5 Conclusion

We revisit high-resolution GUI grounding through the lens of uncertainty. Instead of introducing additional supervision or complex training strategies, AutoFocus leverages token-level perplexity as an intrinsic signal to guide adaptive visual refinement. By converting generative uncertainty into structured spatial modeling, our approach enables principled test-time search without additional training. The consistent gains across models and benchmarks indicate that internal generative uncertainty can serve as a practical foundation for scalable visual reasoning beyond single-pass inference.

LIMITATIONS

AutoFocus improves GUI grounding by allocating additional test-time computation to uncertain examples, but this also introduces extra VLM calls for verification, zoomed inference, and visual aggregation. Although we use a verifier and aggregator of similar model size to the base predictor to avoid introducing substantially stronger model capacity, the resulting pipeline is still more computationally expensive than strictly single-pass baselines. This may limit its applicability in latency-sensitive or resource-constrained GUI-agent settings. In addition, AutoFocus relies on token-level likelihoods or perplexity estimates from autoregressive coordinate generation, which are readily available in many open-source VLM pipelines but may not be exposed by some closed-source APIs. The method may also inherit the biases of the underlying VLM: when the verifier and aggregator fail to recognize an incorrect marked location, the refinement process may not recover from the initial error. Future work could investigate shared-backbone implementations, lightweight verification modules, and broader evaluations on dynamic interfaces and long-horizon GUI tasks.

AI USE STATEMENT

We did not use AI assistance for writing.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Wenjing Hu, Yuchen Mao, et al. Spider2-v: How far are multimodal agents from automating data science and engineering workflows? *Advances in Neural Information Processing Systems*, 37:107703–107744, 2024.

Gongwei Chen, Xurui Zhou, Rui Shao, Yibo Lyu, Kaiwen Zhou, Shuai Wang, Wentao Li, Yinchuan Li, Zhongang Qi, and Liqiang Nie. Less is more: Empowering gui agent with context-aware simplification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5901–5911, 2025a.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, et al. Guicourse: From general vision language model to versatile gui agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21936–21959, 2025b.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 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. *arXiv preprint arXiv:2410.05243*, 2024.

Boyu Gou, Demi Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 30851–30883, 2025.

Zhangxuan Gu, Zhengwen Zeng, Zhenyu Xu, Xingran Zhou, Shuheng Shen, Yunfei Liu, Beitong Zhou, Changhua Meng, Tianyu Xia, Weizhi Chen, et al. Ui-venus technical report: Building high-performance ui agents with rft. *arXiv preprint arXiv:2508.10833*, 2025.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14953–14962, 2023.

Zhiyuan Huang, Ziming Cheng, Junting Pan, Zhaohui Hou, and Mingjie Zhan. Spiritsight agent: Advanced gui agent with one look. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29490–29500, 2025.

Zhiyuan Jiang, Shenghao Xie, Wenyi Li, Wenqiang Zu, Peihang Li, Jiahao Qiu, Siqi Pei, Lei Ma, Tiejun Huang, Mengdi Wang, et al. Zoom in, click out: Unlocking and evaluating the potential of zooming for gui grounding. *arXiv preprint arXiv:2512.05941*, 2025.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Bin Lei, Nuo Xu, Ali Payani, Mingyi Hong, Chunhua Liao, Yu Cao, and Caiwen Ding. Gui-spotlight: Adaptive iterative focus refinement for enhanced gui visual grounding. *arXiv preprint arXiv:2510.04039*, 2025.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9098–9108, 2025a.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025b.

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

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xavier Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, et al. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. *arXiv preprint arXiv:2508.05731*, 2025b.

Tiange Luo, Lajanugen Logeswaran, Justin Johnson, and Honglak Lee. Visual test-time scaling for gui agent grounding. *arXiv preprint arXiv:2505.00684*, 2025.

Mingyu Ouyang, Kevin Qinghong Lin, Mike Zheng Shou, and Hwee Tou Ng. Focusui: Efficient ui grounding via position-preserving visual token selection. *arXiv preprint arXiv:2601.03928*, 2026.

Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Hassan. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6300–6323, 2025.

Joonhyung Park, Peng Tang, Sagnik Das, Srikar Appalaraju, Kunwar Yashraj Singh, R Manmatha, and Shabnam Ghadar. R-vlm: Region-aware vision language model for precise gui grounding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9669–9685, 2025.

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.

Bytedance Seed. Seed2. 0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

Yuchen Sun, Shanhui Zhao, Tao Yu, Hao Wen, Samith Va, Mengwei Xu, Yuanchun Li, and Chongyang Zhang. Gui-xplore: Empowering generalizable gui agents with one exploration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19477–19486, 2025.

Fei Tang, Bofan Chen, Zhengxi Lu, Tongbo Chen, Songqin Nong, Tao Jiang, Wenhao Xu, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Ui-zoomer: Uncertainty-driven adaptive zoom-in for gui grounding. *arXiv preprint arXiv:2604.14113*, 2026a.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33214–33222, 2026b.

Venus Team, Zhuohan Cai, Haoxing Chen, Jiaxuan Chen, Weizhi Chen, Changlong Gao, Zhangxuan Gu, Yuan Guo, Yusong Hu, Jianrong Jiang, et al. Ui-venus-2 technical report. *arXiv preprint arXiv:2609.00028*, 2026.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710, 2024.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, et al. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26257–26267, 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. *arXiv preprint arXiv:2506.03143*, 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: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

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. *arXiv preprint arXiv:2505.13227*, 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. *arXiv preprint arXiv:2412.04454*, 2024.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, et al. Gta1: Gui test-time scaling agent. *arXiv preprint arXiv:2507.05791*, 2025a.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-ui: Visual grounding for gui instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22418–22433, 2025b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 355–385, 2024.

Xianhang Ye, Yiqing Li, Wei Dai, Miancan Liu, Ziyuan Chen, Zhangye Han, Hongbo Min, Jinkui Ren, Xiantao Zhang, Wen Yang, et al. Gui-arp: Enhancing grounding with adaptive region perception for gui agents. *arXiv preprint arXiv:2509.15532*, 2025.

Wenwen Yu, Zhibo Yang, Jianqiang Wan, Sibo Song, Jun Tang, Wenqing Cheng, Yuliang Liu, and Xiang Bai. Omniparser v2: Structured-points-of-thought for unified visual text parsing and its generality to multimodal large language models. *arXiv preprint arXiv:2502.16161*, 2025.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. SE-GUI: Enhancing visual grounding for GUI agents via self-evolutionary reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, et al. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*, 2025b.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Mllms know where to look: Training-free perception of small visual details with multimodal llms. *arXiv preprint arXiv:2502.17422*, 2025.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning*, pp. 61349–61385. PMLR, 2024.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. *arXiv preprint arXiv:2505.15810*, 2025.

A Appendix

A.1 Test-Time Scaling in GUI Agents

Test-time scaling involves scaling computational resources during inference to enhance model performance Wang et al. (2023); Yao et al. (2023); Ye et al. (2024). This approach allows AI agents to allocate additional processing power to challenging scenarios, thereby improving perception and reasoning. Inspired by advancements in textual test-time scaling for LLMs (e.g., Chain-of-thought prompting Wei et al. (2022)), recent studies have extended this logic to the visual domain, where reasoning is structured as a sequential trajectory of continuously adjusted visual content Gupta & Kembhavi (2023); Zheng et al. (2025); Su et al. (2025); Wei et al. (2026). For instance, Visual Programming Gupta & Kembhavi (2023) utilizes auxiliary spatial cues, such as lines drawn on diagrams, to ground visual reasoning. Techniques like Vicrop Zhang et al. (2025) employ a model's internal attention maps and gradient saliency to automate visual cropping, while DYFO Li et al. (2025a) facilitates a bidirectional interaction between LMMs and external "visual experts" using Monte Carlo Tree Search to simulate human-centric focus adjustments. In the specific context of

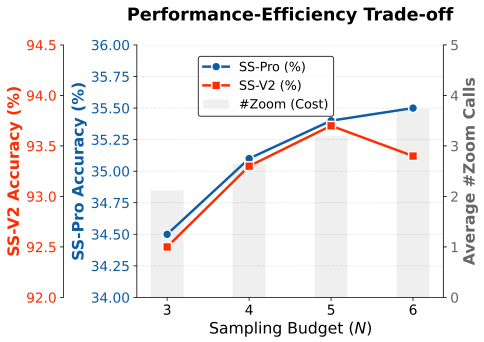


Figure 4: Effect of Sampling Budget N.

GUI agents, Region-Focus Luo et al. (2025) introduces an "image-as-map" mechanism paired with simplified region selection to improve interactivity. Similarly, Dimo-GUI Wu et al. (2025a) dynamically refines grounding results by zooming into sub-regions centered on the model's initial and noisy coordinate predictions. Unlike existing approaches that rely on rigid crops or computationally expensive search algorithms, our framework leverages internal model perplexity as an intrinsic navigation signal, dynamically adapts its search area to transcend the limitations of fixed-region zooming, ensuring high-precision localization even within dense and complex digital interfaces.

A.2 Hyperparameter settings.

We set the sampling temperature to $\tau = 0.75$ and top-p=1. For region proposal generation, we retain $K_{\text{local}} = 3$ local proposals and $K_{\text{global}} = 2$ global proposals. These hyperparameters, including β and α, are empirically determined based on the model's intrinsic perplexity (PPL) distribution and the layout density of different datasets. For most models, including Qwen-series, UI-TARS and UI-Venus, we use $\beta = 50$. For GTA1-series models, which exhibit systematically lower perplexity and sharper confidence distributions, we use a larger value $\beta = 80$ to maintain a comparable exploration range. For global proposal construction, we use scale factors $\alpha \in \{5, 8\}$ to capture uncertainty at multiple spatial extents. We use pink stars as visual markers. The squareness factor is set to $\lambda = 0.5$ to balance spatial precision and contextual coverage. Finally, all region proposals are constrained to have a minimum crop size of 336×336 pixels to ensure sufficient visual detail for reliable refinement.

A.3 Hyperparameter ablations (How sensitive?).

Figure 4 analyzes the effect of the stochastic sampling budget N on performance and computational cost (e.g., average zoom calls). As N increases from 3 to 5, accuracy on ScreenSpot-Pro steadily improves (34.5% $\rightarrow$ 35.4%), while ScreenSpot-V2 shows similar gains (92.5% $\rightarrow$ 93.7%). This confirms that multiple hypothesis sampling enhances spatial uncertainty estimation and proposal diversity. However, increasing N beyond 5 yields only marginal improvements (+0.1% on SS-Pro) while noticeably increasing the average number of zoom operations. This indicates diminishing returns when excessive hypotheses introduce redundant spatial regions. We therefore adopt $N = 5$ as a balanced trade-off between accuracy and inference efficiency.

Squareness factor λ. Table 5 evaluates the sensitivity to the Shape-Aware Zooming parameter λ, which interpolates between anisotropic crops ($\lambda = 0$) and square crops ($\lambda = 1$). When $\lambda = 0$, the model relies purely on uncertainty-derived anisotropic regions, resulting in suboptimal accuracy (33.8% on SS-Pro), likely due to insufficient contextual coverage. Moderate values improve performance, with $\lambda = 0.5$ achieving the best overall results on both benchmarks. Larger values (e.g., $\lambda = 0.75$) slightly

Table 5: Effect of squareness factor λ.

λ	SS-Pro ↑	SS-V2 ↑
0	33.8	92.4
0.25	34.4	93.0
0.5	**35.4**	**93.7**
0.75	35.3	93.5

Table 6: Budget-comparable comparison on ScreenSpot-Pro with Qwen2.5-VL-7B (N=5 for all variants). AutoFocus more than doubles the average gain of pure sample-and-aggregate strategies.

Method (Qwen2.5-VL-7B)	Text	Icon	Avg
Baseline (single pass)	39.3	6.6	26.8
+ Best-of-N (N=5)	44.3	8.3	30.5
+ Self-Consistency (N=5)	43.8	8.0	30.1
+ AutoFocus (Ours)	**50.2**	**11.3**	**35.4**

degrade performance, suggesting that overly square
crops introduce unnecessary background regions and dilute high-resolution detail. These results validate the importance of balancing structural precision and contextual completeness, and demonstrate that AutoFocus is relatively robust within a reasonable λ range.

A.4 Comparison with Test-Time Scaling Baselines

A natural concern for any test-time refinement framework is whether the observed improvements arise from the proposed design itself or simply from the additional inference compute. To disentangle these two factors, we compare AutoFocus against two widely used test-time scaling strategies under *comparable inference budgets*, all built on the same Qwen2.5-VL-7B backbone and evaluated on ScreenSpot-Pro.

Setup. Each baseline performs N=5 stochastic samples, roughly matching the sampling footprint of AutoFocus (N=5 candidates + verifier + at most K_{global}+K_{local} zoomed inferences + visual aggregation):

- **Best-of-N (BoN).** Sample N coordinates with temperature τ=0.75 and keep the one with the lowest sentence-level perplexity (i.e., highest model confidence).
- **Self-Consistency (SC).** Sample N coordinates and aggregate them via spatial majority voting: candidates are clustered with a small radius threshold and the centroid of the largest cluster is returned.
- **AutoFocus (Ours).** The full pipeline with uncertainty-aware Gaussian Dynamic Focusing, Shape-Aware Zooming, and visual aggregation.

Analysis. As shown in Table 6, naively spending additional compute on stochastic sampling yields only limited benefits: BoN and SC improve the baseline by $+3.7$ and $+3.3$ on average, with gains concentrated on text targets and remaining marginal on icon grounding, which is the regime most affected by the high-resolution bottleneck. In contrast, AutoFocus improves the baseline by $+8.6$ on average ($+10.9$ text, $+4.7$ icon), more than doubling the gain of either sampling-only baseline under the same budget. The disproportionately larger improvement on icon targets ($+4.7$ vs. $+1.7$) indicates that the benefit comes from *where* the model looks rather than *how many times* it guesses, confirming that AutoFocus's uncertainty-guided spatial refinement is not reducible to additional test-time sampling.

A.5 Head-to-Head Comparison with Zoom-Based Refinement Methods

To further isolate the contribution of AutoFocus from prior zoom-based refinement strategies, we conduct a head-to-head comparison against the two most representative methods in this line of work, RegionFocus Luo et al. (2025) and Dimo-GUI Wu et al. (2025a), all built on the same Qwen2.5-VL-72B backbone and evaluated on ScreenSpot-Pro. This setting deliberately selects a strong base model so that the remaining gains can only be attributed to the refinement strategy rather than to backbone capacity.

Setup. All three methods follow a coarse-to-fine pipeline that performs an initial prediction followed by zoom-in refinement, but differ in how the zoom region is determined:

Table 7: Head-to-head comparison of zoom-based refinement methods on ScreenSpot-Pro with Qwen2.5-VL-72B. AutoFocus achieves the best results on both Text and Icon, surpassing the strongest prior method by $+3.5$ on average.

Method (Qwen2.5-VL-72B)	Text	Icon	Avg
Baseline (single pass)	64.9	20.2	47.8
+ Dimo-GUI Wu et al. (2025a)	76.8	31.5	59.4
+ RegionFocus Luo et al. (2025)	78.6	34.1	61.6
+ AutoFocus (Ours)	**80.9**	**39.5**	**65.1**

- **RegionFocus** relies on an "image-as-map" mechanism with simplified region selection driven by predefined anchors.
- **Dimo-GUI** zooms into sub-regions centered on the model's initial coordinate prediction with modality-aware visual reasoning.
- **AutoFocus (Ours)** converts axial perplexity into an anisotropic Gaussian field to generate uncertainty-aware global and local proposals, with an error-triggered verifier that activates refinement only on uncertain cases.

Analysis. Table 7 shows that all three refinement strategies substantially improve over the single-pass baseline, confirming the value of coarse-to-fine inference for high-resolution GUI grounding. However, their effectiveness differs notably. Dimo-GUI, which zooms around the initial prediction, gains $+11.6$ on average but is bounded by the quality of that single anchor, leaving icon accuracy at 31.5. RegionFocus further pushes icon performance to 34.1 through saliency-driven region selection, yielding a stronger overall score of 61.6. AutoFocus reaches 65.1 ($+3.5$ over RegionFocus and $+5.7$ over Dimo-GUI), with the largest margin again on icon grounding ($+4.5$ over RegionFocus, $+7.1$ over Dimo-GUI). This pattern suggests that uncertainty-aware spatial reasoning is more reliable than fixed-anchor or saliency-based heuristics, especially for small interactive elements where a single deterministic prediction is most likely to fail.

A.6 Data sources and licensing.

All experiments in this work are conducted exclusively on publicly available GUI grounding benchmarks, namely ScreenSpot-V2 Wu et al. (2024), ScreenSpot-Pro Li et al. (2025b), and OSWorld-G Xie et al. (2025). These datasets are released by their original authors for academic research under their respective licenses, and we strictly follow the intended use specified in each release. Specifically, ScreenSpot-V2 and OSWorld-G are distributed under the Apache-2.0 license, while ScreenSpot-Pro is released under the MIT license. We did not collect, crawl, or annotate any new GUI screenshots, and no human subjects were involved in the construction or evaluation of our framework. To the best of our knowledge, the screenshots included in these benchmarks have been curated by the original authors to avoid personally identifiable information (PII), and we use them *as-is* without any modification, redistribution, or de-anonymization attempt.

A.7 Comparison with Training-Based Uncertainty Methods

We compare AutoFocus with two representative training-based uncertainty methods: GUI-RCPO Du et al. (2026), which estimates implicit confidence from prediction agreement, and GUI-G^2 Tang et al. (2026b), which formulates GUI grounding as a reward modeling problem with Gaussian distributions. Both methods require additional training, whereas AutoFocus operates entirely at test time without parameter updates.

As shown in Table 8, AutoFocus achieves 35.4% with Qwen2.5-VL-7B, outperforming GUI-RCPO (25.93%) by +9.5% despite requiring no training. While GUI-G^2 achieves higher accuracy (47.50%), it requires substantial reinforcement learning with Gaussian reward modeling. AutoFocus scales effectively to larger backbones, reaching 65.1% with Qwen2.5-VL-72B, demonstrating that test-time uncertainty refinement can complement or even substitute for training-based approaches in resource-constrained scenarios.

Table 8: Comparison with training-based uncertainty methods on ScreenSpot-Pro. AutoFocus achieves competitive performance without requiring additional training, and scales effectively to larger backbones.

Method	Base model	Requires training	SS-Pro Avg. (%)
GUI-G^2	Qwen2.5-VL-7B	Yes	47.50
GUI-RCPO	Qwen2.5-VL-7B	No	25.93
GUI-RCPO	GUI-G^2	No	47.88
AutoFocus	**Qwen2.5-VL-7B**	**No**	**35.40**
AutoFocus	**Qwen2.5-VL-72B**	**No**	**65.10**

A.8 Inference Cost Analysis

We measure wall-clock latency to understand the practical accuracy-efficiency trade-off. Experiments are conducted on ScreenSpot-Pro using Qwen2.5-VL-7B deployed as a service with vLLM 0.18.0 on one NVIDIA A100 GPU. We set the maximum number of image tokens to 16,384 and use identical serving, image-processing, and decoding configurations across all settings.

Table 9: Inference time and accuracy trade-off on ScreenSpot-Pro with Qwen2.5-VL-7B. The verifier selectively activates refinement, keeping practical runtime below worst-case cost.

Method	Sampling budget N	Avg. zoom calls	Accuracy (%)	Approx. seconds/example
Qwen2.5-VL-7B	-	0	26.8	3.26
+ AutoFocus	3	2.1	34.5	11.80
+ AutoFocus	4	2.6	35.1	13.43
+ AutoFocus	**5 (default)**	**3.1**	**35.4**	**14.91**
+ AutoFocus	6	3.6	35.5	16.01

Table 9 shows that the average number of zoom calls increases gradually from 2.1 to 3.6 as the sampling budget grows, while end-to-end runtime remains approximately 3.6×–4.9× that of the single-pass baseline. The additional computation primarily comes from sampling multiple initial coordinate hypotheses and performing grounding on proposed zoom-in regions. Because the verifier selectively activates these stages only for uncertain predictions, practical runtime remains below the worst-case cost of applying the complete refinement pipeline to every example.

Increasing N from 3 to the default value of 5 improves accuracy from 34.5% to 35.4%, whereas $N = 6$ adds only 0.1 percentage points with approximately 29 additional minutes of total inference time. These results support $N = 5$ as a practical balance between grounding accuracy and inference efficiency.