V2P: Visual Attention Calibration for GUI Grounding via Background Suppression and Center Peaking

Anonymous ACL submission

Abstract

Precise localization of GUI elements is crucial for the development of GUI agents. Traditional methods rely on bounding box or center-point regression, neglecting spatial interaction uncertainty and visual-semantic hierarchies. Recent methods incorporate attention mechanisms but still face two key issues: (1) ignoring processing background regions causes attention drift from the desired area, and (2) uniform modeling the target UI element fails to distinguish between its center and edges, leading to click imprecision. Inspired by how humans visually process and interact with GUI elements, we propose the Valley-to-Peak (V2P) method to address these issues. To mitigate background distractions, V2P introduces a suppression attention mechanism that minimizes the model's focus on irrelevant regions to highlight the intended region. For the issue of center-edge distinction, V2P applies a Fitts' Law-inspired approach by modeling GUI interactions as 2D Gaussian heatmaps where the weight gradually decreases from the center towards the edges. The weight distribution follows a Gaussian function, with the variance determined by the target's size. Consequently, V2P effectively isolates the target area and teaches the model to concentrate on the most essential point of the UI element. The model trained by V2P achieves the performance with 92.4% and 52.5% on two benchmarks ScreenSpot-v2 and ScreenSpot-Pro. Ablations further confirm each component's contribution, underscoring V2P's generalizability in precise GUI grounding tasks and its potential for real-world deployment in future GUI agents.

1 Introduction

Recent advances in large language models (LLMs) and vision-language models (VLMs) have enabled agents to interpret natural language instructions and interact with graphical user interfaces (GUIs) across desktop, mobile, and web platforms. Central to this capability is GUI grounding, which aligns language commands with semantically relevant UI elements and their spatial locations (Cheng et al., 2024). This task bridges user intent and interface actions, supporting the development of intelligent, general-purpose agents for real-world human-computer interaction.

Early approaches framed GUI grounding as coordinate generation task, outputting a bounding box or (x, y) coordinate for a natural-language query (Zhang et al., 2025; Qin et al., 2025). However, this "coordinate generation" method suffers weak spatial–semantic alignment (Wu et al., 2025), treating coordinates like ordinary words without inherent spatial meaning. Moreover, point-wise regression contradicts the multi-point validity inherent in real interactions. Recent work addresses these issues by leveraging the model's attention maps (Wu et al., 2025). Instead of predicting coordinates, it extracts cross-modal attention weights linking instruction tokens to image patches, selecting the most attended patch as the click position. This approach offers dense spatial supervision and naturally tolerates multiple valid click regions, aligning better with human behavior.

However, after manually scrutinizing the attention heatmap of these methods mentioned above, we found two main issues, as shown in Fig. 1:

1. **Background Distraction**: Current loss functions only reward attention on target patches but fail to penalize it on the background. This leads to a "divergent" attention distribution where background regions also receive high scores. Consequently, softmax normalization allows these regions to absorb probability mass, weakening or even shifting the intended attention peak.

2. **Centre-edge Confusion**: Because labels treat all pixels within a bounding box equally, the model cannot differentiate an element's center from its edges, resulting in uniform attention

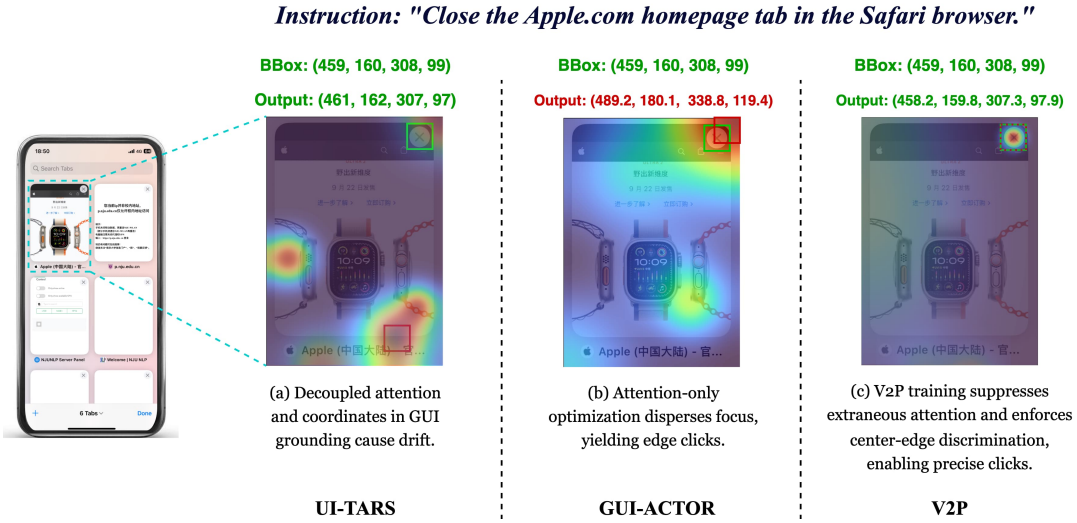


Figure 1: Comparison of different strategies in the GUI grounding task. The green box marks the ground-truth bounding box, and the red box highlights the region where the model places the highest attention given the instruction and screenshot. The overlaid heatmap is colour-coded from cool (blue) to warm (red), with warmer colours indicating higher attention values.

and inaccurate clicks that miss the center. Furthermore, for small elements, this often leads the attention to drift towards the edges, making the model more prone to mislocalization, especially when elements overlap.

This raises a key question: *How can we guide the model's attention to focus more precisely on the target UI element?* Motivated by human behavior—first isolating the target (valley suppression) then focusing on the action point (peak emphasis)—we propose **Valley-to-Peak (V2P)**. V2P suppresses distractions by creating low-attention "valleys" in irrelevant areas while sharpening a "peak" at the actionable center.

Suppression Attention: We apply inverse attention regularization (Li et al., 2018) to penalize high attention outside the target, isolating true UI elements and reducing attention to non-target regions.

Fitts-Gaussian Peak Modeling: Inspired by Fitts' Law (MacKenzie, 1992; Fitts, 1954), we use a 2D Gaussian centered on the target, scaled to its size, to model human's click likelihood, which yields a heatmap that peaks at the center and decays towards the edges, better matching real user interactions.

Together, these modules reshape the attention map, enhancing grounding precision by aligning the model's focus with human patterns.

Our contribution can be summarized as follows:

1. We systematically analyze existing attention-based methods for visual grounding in GUI agents and, through statistical evaluation, identify two main issues——*Background Distraction* and *Center-Edge Confusion*. In addition, we provide a detailed analysis of the underlying causes of these issues and provide insights for further improvements.

2. We introduce *Attention Suppression Mechanism (SA)* to mitigate Background Distraction and employ *Fitts-Gaussian Peak Modeling (FGPM)* to effectively alleviate Center-Edge Confusion. Building on these methods, we propose the **Valley-to-Peak (V2P)** framework, an agentic learning paradigm for GUI grounding that significantly enhances the localization precision and accuracy of Vision-Language Models on GUI elements.

3. Extensive experiments demonstrate that V2P achieves advanced performance on multiple public benchmarks, reaching 92.4% on ScreenSpot-v2 and 52.50% on the challenging ScreenSpot-Pro, with relative improvements of 3.6% and 25.7%. Furthermore, we confirm that V2P demonstrates significant practical value for real-world deployment and seamless integration into GUI agents.

2 Related Work

2.1 GUI-Agents

GUI agents have progressed from rudimentary random- or rule-based test tools to multimodal,

LLM-driven systems that can follow natural-language instructions. Early efforts such as Monkey testing (Wetzlmaier et al., 2016) and planning or script record-and-replay frameworks (Memon et al., 2001; Steven et al., 2000) provided basic coverage but required hand-crafted rules or scripts. Machine-learning techniques later enabled more adaptive behaviour: Humanoid (Li et al., 2020) and Deep GUI (YazdaniBanafsheDaragh and Malek, 2022) learned user-like action policies from screenshots, while widget detectors (White et al., 2019) improved element recognition. Natural-language interfaces soon followed, e.g. FLIN (Mazumder and Riva, 2021) and RUSS (Xu et al., 2021), and reinforcement learning environments like WoB (Shi et al., 2017) and WebShop (Yao et al., 2023) pushed web-scale interaction. The recent arrival of LLMs has unified perception, reasoning and control: WebAgent (Gur et al., 2024) and WebGUM (Furuta et al., 2024) achieve open-world browsing, AutoDroid (Wen et al., 2024) and AppAgent (Zhang et al., 2023) automate smartphones, and desktop agents such as UFO (Zhang et al., 2024) demonstrate GPT-4-level capabilities; industrial systems (e.g. Claude 3.5 Sonnet and Operator) further attest to the practical traction of GUI agents.

2.2 GUI Grounding

Prevalent approaches in GUI grounding typically frame the problem as a coordinate generation task (Zhang et al., 2025). Models such as UI-TARS (Qin et al., 2025) and CogAgent (Hong et al., 2024) utilize massive supervised fine-tuning to train VLMs to autoregressively generate textual numerical coordinates to ground the target element. However, treating spatial coordinates as ordinary language tokens can limit fine-grained visual alignment. Consequently, recent methods have largely shifted to leveraging the cross-modal attention maps of Vision-Language Models (VLMs) (Wu et al., 2025). In this paradigm, the model's prediction is derived from the image patch with the highest attention score in response to a language command. While more robust, this approach often suffers from imprecise attention, with focus leaking into irrelevant background regions or spreading too uniformly across the target element. Our work directly addresses this by refining the quality of the attention map itself.

Our approach, V2P, draws inspiration from two distinct areas. To create attention "valleys" and suppress background noise, we adopt attention suppression techniques that penalize focus outside the target region (Li et al., 2018). To form a sharp "peak" at the target's center, we are inspired by both Fitts' Law from Human-Computer Interaction (HCI) (MacKenzie, 1992) and the common practice of using Gaussian heatmaps in localization tasks like pose estimation (Fitts, 1954). To our knowledge, our work is the first to synergistically combine background suppression with center-focused peak modeling to simulate the human pattern of interaction with the UI elements.

3 Method

We introduce Valley-to-Peak (V2P), a method that reshapes the model's attention landscape to mimic human focus patterns for precise GUI grounding. It achieves this through two synergistic components:

- **Suppression Attention Valley Constraint:** Penalizes attention on irrelevant regions to form low-attention "valleys," effectively suppressing background distractions.
- **Fitts-Gaussian Peak Modeling:** Models interaction likelihood with a size-adaptive 2D Gaussian, creating a sharp attention "peak" at the target's most actionable center.

By jointly optimizing these objectives, V2P produces a continuous, spatially-aware attention map that overcomes the limitations of rigid, uniform labels used in prior work. Below, we first outline the overall architecture (Sec. 3.1), then detail the Suppression Attention (Sec. 3.2) and Fitts-Gaussian Peak Modeling (Sec. 3.3) components.

3.1 Model Architecture Overview

We build upon GUI-Actor (Wu et al., 2025), a coordinate-free visual grounding framework that localizes GUI actions through attention rather than coordinate regression. Given a screenshot I and an instruction q, the model introduces a special token <ACTOR> in the output sequence as a contextual anchor. The final-layer hidden state of <ACTOR>, denoted $h_{\texttt{<ACTOR>}}$, is used to compute action attention over image patch features $\{v_1, \ldots, v_M\}$ extracted by the vision encoder.

To enhance spatial coherence among visual patches, we apply a self-attention module over the patch features:

$$\tilde{v}_1, \ldots, \tilde{v}_M = \text{SelfAttn}(v_1, \ldots, v_M) \quad (1)$$

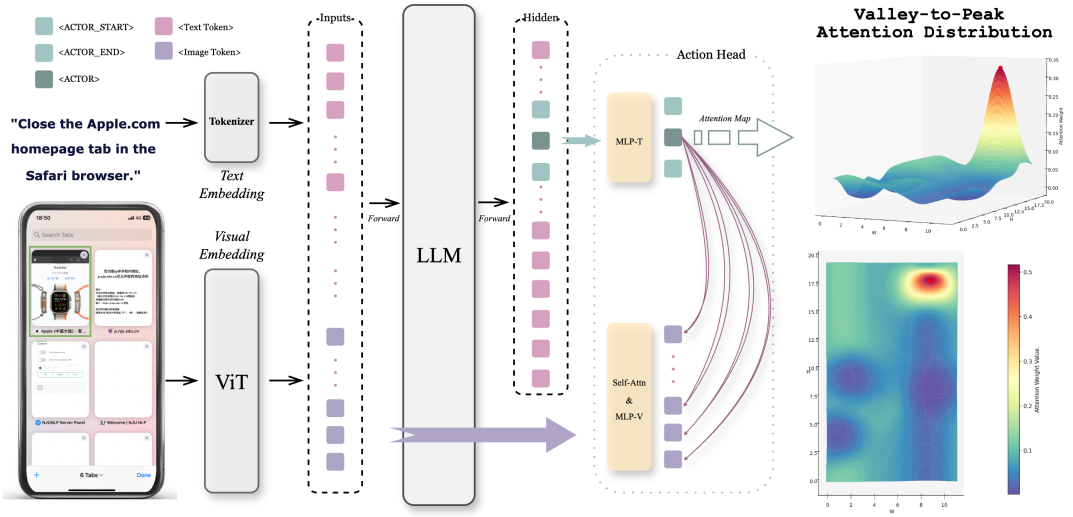


Figure 2: **Valley-to-Peak training method (V2P).** V2P jointly suppresses noise and enhances signals via two strategies: An inverse-attention penalty carves valleys in non-target areas, while size-adaptive Fitts-Gaussian peaks create sharp peaks at UI elements' centers. This dual approach reshapes attention maps (rightmost example), enabling the model to quickly pinpoint interaction points in cluttered interfaces.

yielding contextualized representations. These are projected into a shared embedding space with $h_{\texttt{<ACTOR>}}$ via separate MLPs:

$$z = \text{MLP}_T(h_{\texttt{<ACTOR>}}), \tag{2}$$

$$z_i = \text{MLP}_V(\tilde{v}_i), \quad i = 1, \ldots, M. \tag{3}$$

Attention scores are then computed as:

$$\begin{aligned} \alpha_i &= \frac{z^\top z_i}{\sqrt{d}} \\ a_i &= \frac{\exp(\alpha_i)}{\sum_{j=1}^{M} \exp(\alpha_j)} \end{aligned} \tag{4}$$

where d is the embedding dimension. The resulting $\{a_i\}_{i=1}^{M}$ forms a normalized attention distribution over the M image patches, representing the model's belief about the target interaction location.

3.2 Suppression Attention Constraint for Distraction Mitigation

Attention maps in complex interfaces can suffer from *attention leakage*, where notable responses are mistakenly assigned to regions far from the target area, particularly in the presence of visually similar distracting patches. To address this issue and enhance spatial precision, we propose a Suppression Attention Constraint. This mechanism explicitly penalizes attention allocated to non-target regions, enforcing sparsity and improving the model's ability to distinguish targets from surrounding distractions.

Let $\mathcal{G} \subset \{1, \ldots, M\}$ denote the set of patch indices whose spatial support R_i has empty intersection with the ground-truth bounding box b:

$$\mathcal{G} = \{i \in \{1, \ldots, M\} \mid R_i \cap b = \emptyset\} \tag{5}$$

We define the attention loss as the total attention mass over these irrelevant regions:

$$\mathcal{L}_{\text{Attn}} = \sum_{i \in \mathcal{G}} a_i \tag{6}$$

To better understand the theoretical foundation of this constraint, we analyze the gradient dynamics of attention weights. For the target patch k with attention weight $A_k = \text{softmax}(s_k)$, the gradient with respect to any non-target patch logit s_i is:

$$\begin{aligned} w_i &= \frac{\partial A_k}{\partial s_i} = \frac{\partial \text{softmax}(s_k)}{\partial s_i} \\ &= -\frac{e^{s_k} e^{s_i}}{(\sum_i^M e^{s_i})^2} = -A_k A_i < 0 \quad (i \neq k) \end{aligned} \tag{7}$$

This gradient analysis reveals that any increase in attention logits s_i for non-target patches negatively impacts the target attention A_k. The magnitude $|w_i| = A_k A_i$ quantifies this negative influence: larger values indicate that even small increases in attention to patch i will cause rapid degradation in target attention A_k. This theoretical insight naturally motivates using $|w_i|$ as a weighting factor in our suppression loss, providing stronger

penalties for patches that pose greater threats to target attention focus. And we have the *suppression attention loss* combined with gradient weight as:

$$\mathcal{L}_{\text{Sup\_Attn}} = \sum_{i \in \mathcal{G}} w_i a_i \tag{8}$$

This loss encourages the model to suppress attention on irrelevant regions, thereby reducing the impact of distracting elements in cluttered interfaces. By explicitly minimizing $\mathcal{L}_{\text{Sup\_Attn}}$, the model is incentivized to concentrate its focus on the target region, resulting in enhanced spatial precision and improved robustness.

3.3 Fitts-Gaussian Peak Modeling for Center-Focused Grounding

While the Suppression Attention Constraint encourages focus on target regions, overlapping UI elements can still lead to attention dispersion—particularly toward the boundaries of positively labeled components—resulting in ambiguous and spatially diffused attention maps.

Our supervision strategy is inspired by Fitts' Law (MacKenzie, 1992; Fitts, 1954), which reveals that click probability peaks at the center of an UI element and decays toward its edges, closely following a Gaussian distribution. We encode this behavior with Fitts-Gaussian Peak Modeling to guide the model's focus in line with observed human interaction.

Specifically, we model the ideal attention distribution as a 2D Gaussian density centered at the centroid of the ground-truth bounding box $b = [x_1, y_1, x_2, y_2]$:

$$\mu = (c_x, c_y) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \tag{9}$$

To reflect the interaction tolerance associated with target size, we set the standard deviation of the Gaussian proportional to the element's width and height:

$$\sigma_x = \frac{w}{\sigma_{\text{factor}}}, \quad \sigma_y = \frac{h}{\sigma_{\text{factor}}} \tag{10}$$

where $w = x_2 - x_1$, $h = y_2 - y_1$, and σ_{factor} is a hyperparameter controlling the concentration of the attention prior. This formulation ensures that larger elements—more tolerant to pointing errors—induce broader attention peaks, while smaller elements require sharper focus.

Given an input image partitioned into $M = H \times W$ non-overlapping patches of size $s \times s$, we compute the expected attention mass for each patch i, covering spatial region $R_i = [x^i_{\min}, x^i_{\max}] \times [y^i_{\min}, y^i_{\max}]$, by integrating the 2D Gaussian density over R_i:

$$y_i = \int_{R_i} \mathcal{N}(x, y; \mu, \Sigma) dx\, dy \tag{11}$$

where $\Sigma = \text{diag}(\sigma_x^2, \sigma_y^2)$. Thanks to axis-aligned separability, this integral decomposes efficiently into the product of two univariate cumulative distribution functions (CDFs):

$$\begin{aligned} y_i = &\left[\Phi(x^i_{\max}; c_x, \sigma_x) - \Phi(x^i_{\min}; c_x, \sigma_x)\right] \\ &\cdot \left[\Phi(y^i_{\max}; c_y, \sigma_y) - \Phi(y^i_{\min}; c_y, \sigma_y)\right] \end{aligned} \tag{12}$$

with $\Phi(\cdot\,; \mu, \sigma)$ denoting the CDF of a univariate normal distribution.

To supervise the model's predicted attention distribution $\{a_i\}$, we adopt the action attention loss from GUI-Actor (Wu et al., 2025), using the Kullback-Leibler (KL) divergence to measure the discrepancy between the target p and prediction a:

$$\begin{aligned} \mathcal{L}_{\text{Action\_Attn}} &= \sum_{i=1}^{M} p_i \log \frac{p_i}{a_i}, \\ p_i &= \frac{y_i}{\sum_{j=1}^{M} y_j + \epsilon}, \\ & i = 1, \ldots, M \end{aligned} \tag{13}$$

where ϵ is a small constant for numerical stability.

Fitts-Gaussian Peak Modeling establishes a center-biased, size-aware attention prior that closely mimics human pointing behavior. By discouraging boundary leakage and promoting centralized attention in a graded, interaction-informed manner, it enhances localization precision and improves robustness in complex and cluttered UI layouts.

3.4 Valley-to-Peak Training

The overall training objective combines next-token prediction loss with action-focused attention losses:

$$\mathcal{L} = \mathcal{L}_{\text{NTP}} + \lambda_1 \mathcal{L}_{\text{Sup\_Attn}} + \lambda_2 \mathcal{L}_{\text{Action\_Attn}} \tag{14}$$

where $\mathcal{L}_{\text{Sup\_Attn}}$ suppresses attention outside the target region (Section 3.2), and $\mathcal{L}_{\text{Action\_Attn}}$ enforces alignment between predicted attention and a Gaussian-shaped target distribution (Section 3.3).

Minimizing the combined loss supports a *Valley-to-Peak* training paradigm: coarse suppression followed by fine-grained alignment. $\mathcal{L}_{\text{Sup\_Attn}}$ first suppresses distractions, guiding attention toward the target region. Then, $\mathcal{L}_{\text{Action\_Attn}}$ sharpens this focus by prioritizing the target's center. This reduces misclicks and alleviates ambiguity caused by overlapping labels, ensuring precise and human-like attention alignment. The coarse-to-fine control enables robust interaction predictions, even in dense and visually complex UI environments.

4 Experiment

4.1 Experimental Setup

We utilize Qwen2.5-VL-Instruct (both 7B and 3B) (Bai et al., 2025) as backbones. To ensure a rigorously fair comparison and isolate algorithmic contributions, we strictly follow the data recipe of the baseline GUI-Actor (Wu et al., 2025), with a learning rate of 5e-6 and $\sigma = 1.0$. Comprehensive details are provided in App. A.

We evaluate on a comprehensive suite of six benchmarks. Our primary evaluation focuses on *ScreenSpot-v2* (Wu et al., 2024b) and *ScreenSpot-Pro* (Li et al., 2025), as they provide the most standardized assessment across diverse platforms and challenging high-resolution OOD scenarios.

To further verify robustness and agentic potential, we also test on *OSWorld-G* (Xie et al., 2025a), *UI-Vision* (Element Grounding) (Nayak et al., 2025), *UI-I2E* (Liu et al., 2025a), and *MMBench-GUI L2* (Liu et al., 2024).

4.2 Main Results

Tab. 1 presents a comprehensive evaluation of V2P against other baselines.

Superior Performance on ScreenSpot Benchmarks. As our primary evaluation field, V2P-7B demonstrates exceptional capabilities among models of similar scale. On *ScreenSpot-v2*, it achieves a competitive accuracy of 92.4%. More critically, on the high-difficulty *ScreenSpot-Pro*, which features high-resolution screens and OOD applications, V2P-7B attains 52.5%, significantly outperforming the strong baseline GUI-Actor-7B (44.6%) and UI-TARS-72B (38.1%). This substantial margin validates that V2P's attention calibration is particularly effective in handling the dense, visually complex interfaces typical of professional GUI environments.

Generalization to Agentic Scenarios. To assess the model's potential as a perception backend for autonomous agents, we extend our evaluation to four benchmarks featuring interaction traces and functional reasoning requirements: *OSWorld-G* (Xie et al., 2025a), *UI-Vision* (Element Grounding) (Nayak et al., 2025), *UI-I2E* (Liu et al., 2025a), and *MMBench-GUI L2* (Liu et al., 2024). As shown in Tab. 1, V2P-7B demonstrates superior performance across the majority of evaluations. Notably, V2P-7B surpasses all other baselines on *UI-Vision*, *UI-I2E*, and *MMBench-GUI L2*. This consistent superiority highlights the model's exceptional functional reasoning and semantic understanding. Furthermore, on *OSWorld-G*, V2P matches the specialist JEDI-7B (Xie et al., 2025a) (52.5%) despite using only ∼50k PC samples versus JEDI's millions. Moreover, V2P significantly surpasses JEDI on other benchmarks, highlighting superior data efficiency and generalization beyond specific domains.

Scalability and Efficiency. As shown in the *Controlled Comparison* group, V2P-3B consistently outperforms its direct competitor GUI-Actor-3B across all six benchmarks. Notably, on some challenging benchmarks, it even surpasses significantly larger scale models. This result underscores the pure algorithmic superiority of the V2P framework and its consistent effectiveness across varying model scales.

4.3 Ablation and Analysis

4.3.1 Component Ablation Study

To validate the necessity of our proposed modules, we conducted a standard ablation study on *ScreenSpot-Pro* (Tab. 2). Removing *Fitts-Gaussian Peak Modeling (FGPM)* leads to a significant performance drop of 5.0%, confirming its critical role in precise localization. Further removing *Suppression Attention (SA)* results in an additional loss of 3.2%. These results verify that both modules are indispensable for the V2P framework.

4.3.2 Attribution of Performance Gains

To investigate the underlying reasons for V2P's superior performance, we conducted a quantitative performance gains attribution analysis on 182 samples where V2P-7B successfully corrected the failures of the baseline GUI-Actor-7B (Wu et al., 2025). As shown in Tab. 3, the results reveal that 50.5% of the performance gains stem from effectively suppressing *Background Distraction*, while

Model	General Grounding		Complex & Semantic Grounding			
	ScreenSpot-v2	ScreenSpot-Pro	OSWorld-G	UI-Vision	UI-I2E	MMBench
Proprietary & General VLMs						
GPT-4o	80.7	0.8	–	1.38	–	2.87
Operator	70.5	–	40.6	–	–	–
Qwen2.5-VL-3B	80.9	16.1	27.3	–	41.7	–
Qwen2.5-VL-7B	88.8	26.8	31.4	0.85	53.8	33.9
GUI-Specialized Models (SFT)						
SeeClick-9.6B	55.1	1.1	–	5.39	26.4	–
OS-Atlas-7B	84.1	18.9	27.7	9.02	58.6	41.4
Aguvis-7B	86.0	22.9	38.7	13.7	53.2	45.7
UGround-V1-7B	87.6	31.1	36.4	12.9	70.3	65.7
UI-TARS-7B	91.6	35.7	47.5	17.6	61.4	64.3
JEDI-7B	91.7	39.5	54.1	24.8	–	–
UI-TARS-72B	90.3	38.1	**57.1**	25.5	73.7	74.3
Controlled Comparison (Identical Training Data)						
GUI-Actor-3B	91.0	42.2	45.9	21.9	63.7	73.5
V2P-3B (Ours)	**91.4**	**48.5**	**48.8**	**26.0**	**69.5**	**77.6**
GUI-Actor-7B	92.1	44.6	49.3	24.3	68.2	76.5
V2P-7B (Ours)	**92.4**	**52.5**	**52.5**	**28.8**	**75.6**	**79.9**

Table 1: **Main Results Comparison.** We evaluate V2P against state-of-the-art baselines across six diverse benchmarks, covering general, high-resolution, and agentic GUI scenarios. V2P-7B significantly outperforming baselines under comparable settings.

Model Variant	Pro Avg.	Δ
V2P-7B (Full)	**52.5**	-
w/o FGPM	47.5	-5.0
w/o FGPM & SA	44.3	-8.2

Table 2: **Component Ablation on ScreenSpot-Pro.** Both FGPM and SA contribute significantly to the final performance.

35.7% are attributed to resolving *Center-Edge Confusion*. This provides strong empirical evidence that V2P's dual-loss mechanism functions exactly as designed.

Baseline Error Type	Count	Contribution
Background Distraction	92	**50.5%**
Center-Edge Confusion	65	**35.7%**
Other / Normal Attention	25	13.7%
Total Improved Samples	182	100%

Table 3: **Performance Gains Attribution Analysis.** We analyzed samples from ScreenSpot-Pro where V2P-7B made correct predictions while the baseline GUI-Actor (Wu et al., 2025) failed. The majority of gains come from correcting background and center-edge errors.

4.3.3 Performance Leap on Tiny Targets

To evaluate performance on fine-grained targets, we categorized UI elements across *ScreenSpot-v2* and *ScreenSpot-Pro* based on their area relative to the patch size n (14×14). Specifically, elements are classified as Small ($n \leq A < 4n$), Medium ($4n \leq A < 9n$), and Large ($A \geq 9n$). As shown in Tab. 4, V2P-7B outperforms the baseline GUI-Actor-7B (Wu et al., 2025) by 10.0% on these small elements. This demonstrates the superiority of V2P in the fine-grained positioning of small targets.

Furthermore, the data shown in Tab. 4 also reveals a critical distribution shift between benchmarks: *ScreenSpot-v2* is dominated by large elements (size $> 9n$), which offer vast spatial tolerance. Consequently, even spatially diffuse attention maps often fall within these generous boundaries, which explains the high accuracy of the baseline on *ScreenSpot-v2*, effectively masking its inherent localization imprecision. In contrast, *ScreenSpot-Pro* is densely populated with small elements that tolerate negligible error. Consequently, V2P-7B's precision advantage, while masked on the coarse-grained *ScreenSpot-v2*, is fully realized on the challenging *ScreenSpot-Pro*.

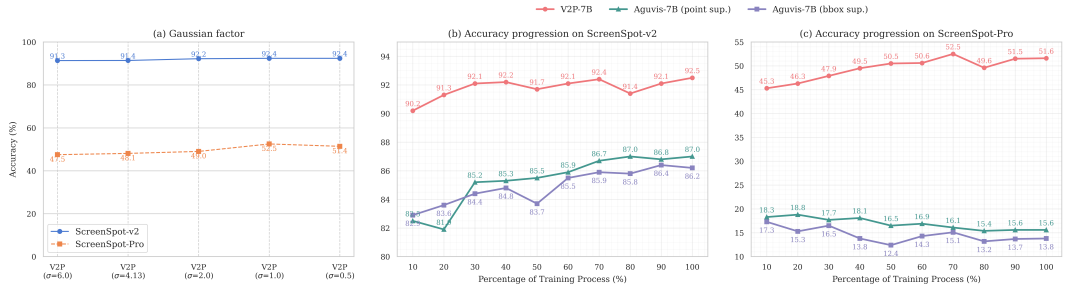


Figure 3: **Gaussian Factor and Generalization Ability Analysis.** (a) Impact of Gaussian Factor σ. A smaller σ (sharper peak) benefits precision, with the optimal performance achieved at $\sigma = 1.0$ for ScreenSpot-Pro. Larger σ values degrade performance due to introduced label noise. (b, c) Generalization Ability. V2P shows consistent improvement, whereas the baseline suffers from overfitting on OOD data.

4.3.4 Sensitivity to Gaussian Factor σ

To analyze the impact of the Gaussian factor σ on grounding precision, we conducted ablation experiments on *ScreenSpot-v2* and *ScreenSpot-Pro* across varying σ values. As shown in Fig. 3(a), model performance is strongly sensitive to this hyperparameter. On *ScreenSpot-v2*, accuracy improves from 91.3% ($\sigma = 6.0$) to 92.4% ($\sigma = 0.5$). Similarly, *ScreenSpot-Pro* achieves its peak accuracy of 52.5% at $\sigma = 1.0$, while larger σ values cause a significant decline.

Element Size	ScreenSpot-v2		ScreenSpot-Pro	
	GUI-Actor	*V2P*	*GUI-Actor*	V2P
Small ($n \sim 4n$)	50.0%	**60.0%**	17.5%	**23.8%**
Medium ($4n \sim 9n$)	71.4%	**85.7%**	43.1%	**47.9%**
Large ($> 9n$)	93.2%	92.9%	60.3%	**66.6%**

Table 4: **Size-stratified Performance.** V2P achieves substantial gains on small elements in ScreenSpot-v2 and ScreenSpot-Pro, underscoring its superior capability in precise fine-grained localization.

We attribute this phenomenon to the spatial concentration of the attention mechanism. Larger σ values generate broader Gaussian distributions, which tend to dilute the spatial focus and introduce background noise into the attention maps. Conversely, a smaller σ produces sharper Gaussian peaks. This acts as a tight spatial constraint, allowing the model to localize UI elements with higher precision and resulting in more accurate click predictions. These results underscore the necessity of balancing σ: while excessively large values hinder localization, a moderately small σ (e.g., 1.0) significantly enhances spatial accuracy.

4.3.5 Training Stability and Generalization

Finally, we evaluate the training stability of V2P-7B compared to the Aguvis-7B (Xu et al., 2025a). As visualized in Fig. 3(b) and (c), V2P-7B demonstrates a consistently ascending accuracy curve on both in-distribution (*ScreenSpot-v2*) and out-of-distribution (*ScreenSpot-Pro*) benchmarks. In sharp contrast, Aguvis-7B (Xu et al., 2025a) exhibits a distinct "overfitting-to-distribution" pattern: while its performance improves on *ScreenSpot-v2*, it suffers from a continuous performance decline on the OOD *ScreenSpot-Pro* after the 20% training milestone. This confirms that our human-like visual attention mechanism (*Fitts-Gaussian Peak Modeling* and *Suppression Attention*) effectively mitigates the overfitting inherent to textual coordinate supervision, ensuring robust generalization across unseen scenarios.

5 Conclusion

In this paper, we address the critical bottlenecks of *Background Distraction* and *Center-Edge Confusion* in GUI grounding by proposing **Valley-to-Peak (V2P)** framework. Mimicking human visual processing, V2P synergizes *Suppression Attention* to eliminate background noise and *Fitts-Gaussian Peak Modeling* to construct sharp, size-adaptive peaks at actionable centers.

By emulating this human-like strategy for visual localization, our approach fosters a more authentic spatial understanding of complex interfaces. Extensive experiments confirm the effectiveness of this framework: V2P achieves exceptional results on ScreenSpot-v2 (92.4%) and the challenging ScreenSpot-Pro (52.5%), consistently outperforming existing strong baselines. Notably, our method demonstrates remarkable robustness on fine-grained small targets and out-of-distribution scenarios, effectively bridging the gap between coarse perception and precise actuation. By enabling agents to "see" and "focus" like human users, V2P offers a scalable and robust foundation for the next generation of general-purpose GUI agents.

Limitations

While V2P demonstrates exceptional performance across various benchmarks, several limitations remain to be addressed:

- **Ambiguity among Semantically Similar Targets:** As analyzed in our failure case studies (see App. D), the model occasionally struggles when multiple UI elements share high semantic similarity, such as identical icons with different functional purposes. This suggests that visual calibration alone may not fully resolve deep logical intent without more comprehensive UI context.
- **Generalization to Unconventional Designs:** The model's attention distribution can become highly dispersed when encountering unconventional or cluttered layouts that deviate from the training distribution, indicating uncertainty in complex visual environments.
- **Computational Overhead:** The introduction of the self-attention module to enhance spatial coherence among visual patches may introduce marginal increases in inference latency compared to simple coordinate regression methods, particularly when processing high-resolution screenshots with a large number of patches.

Ethics Statement

In this work, we propose the Valley-to-Peak (V2P) framework to improve GUI grounding by mimicking human visual processing. We adhere to the ACL Code of Ethics and highlight the following:

- **Data Privacy:** All training and evaluation datasets used in this study are from publicly available academic sources. We have strictly followed data recipe guidelines to exclude samples containing personal identifiable information (PII).
- **Mitigation of Bias:** Our training data spans multiple operating systems and platforms (Mobile, Desktop, Web) to minimize algorithmic bias toward specific UI design patterns.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful feedback. We acknowledge the use of generative AI tools for polishing the linguistic quality and refining the prose of this manuscript. All technical claims and final content remain the sole responsibility of the authors.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Shuai Ren, and Hongsheng Li. 2025. Amex: Android multi-annotation expo dataset for mobile gui agents. *Preprint*, arXiv:2407.17490.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2025. Guicourse: From general vision language models to versatile gui agents. *Preprint*, arXiv:2406.11317.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.

Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47 6:381–91.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2024. Multimodal web navigation with instruction-finetuned foundation models. *Preprint*, arXiv:2305.11854.

Google. 2024. Claude 3.5 sonnet model card addendum. In *Claude 3.5 Sonnet Model Card Addendum*.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025a. Navigating the digital world as humans do: Universal visual grounding for gui agents. *Preprint*, arXiv:2410.05243.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025b. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *The Thirteenth International Conference on Learning Representations*.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A real-world webagent with planning, long context understanding, and program synthesis. *Preprint*, arXiv:2307.12856.

Wenyi Hong, Weihan 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.

Kaixin Li, Meng Ziyang, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. Screenspot-pro: GUI grounding for professional high-resolution computer use. In *Workshop on Reasoning and Planning for Large Language Models*.

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. 2018. Tell me where to look: Guided attention inference network. *Preprint*, arXiv:1802.10171.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024. On the effects of data scale on ui control agents. *Preprint*, arXiv:2406.03679.

Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. 2020. Humanoid: A deep learning-based approach to automated black-box android app testing. *Preprint*, arXiv:1901.02633.

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.

Xinyi Liu, Xiaoyi Zhang, Ziyun Zhang, and Yan Lu. 2025a. Ui-e2i-synth: Advancing gui grounding with large-scale instruction synthesis. *Preprint*, arXiv:2504.11257.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

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

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. 2025. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *Preprint*, arXiv:2503.21620.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. 2025. Gui-r1 : A generalist r1-style vision-language action model for gui agents. *Preprint*, arXiv:2504.10458.

I. Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139.

Sahisnu Mazumder and Oriana Riva. 2021. Flin: A flexible natural language interface for web navigation. *Preprint*, arXiv:2010.12844.

A.M. Memon, M.E. Pollack, and M.L. Soffa. 2001. Hierarchical gui test case generation using automated planning. *IEEE Transactions on Software Engineering*, 27(2):144–155.

Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A. Rodriguez, Montek Kalsi, Rabiul Awal, Nicolas Chapados, M. Tamer Özsu, Aishwarya Agrawal, David Vazquez, Christopher Pal, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. 2025. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. *Preprint*, arXiv:2503.15661.

OpenAI. 2023. OpenAI Operator. Accessed: 2023-10-13.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, and 16 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *Preprint*, arXiv:2501.12326.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR.

John Steven, Pravir Chandra, Bob Fleck, and Andy Podgurski. 2000. jrapture: A capture/replay tool for observation-based testing. *SIGSOFT Softw. Eng. Notes*, 25(5):158–167.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025a. Gui-g^2: Gaussian reward modeling for gui grounding. *Preprint*, arXiv:2507.15846.

Jiaqi Tang, Yu Xia, Yi-Feng Wu, Yuwei Hu, Yuhui Chen, Qing-Guo Chen, Xiaogang Xu, Xiangyu Wu, Hao Lu, Yanqing Ma, Shiyin Lu, and Qifeng Chen. 2025b. Lpo: Towards accurate gui agent interaction via location preference optimization. *Preprint*, arXiv:2506.09373.

Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. Omniparser: A unified framework for text spotting, key information extraction and table recognition. *Preprint*, arXiv:2403.19128.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024. Autodroid: Llm-powered task automation in android. *Preprint*, arXiv:2308.15272.

Thomas Wetzlmaier, Rudolf Ramler, and Werner Putschögl. 2016. A framework for monkey gui testing. In *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 416–423.

Thomas D. White, Gordon Fraser, and Guy J. Brown. 2019. Improving random gui testing with image-based widget detection. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, page 307–317, New York, NY, USA. Association for Computing Machinery.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, and 1 others. 2025. Gui-actor: Coordinate-free visual grounding for gui agents. *arXiv preprint arXiv:2506.03143*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. 2024a. Os-atlas: A foundation action model for generalist gui agents. *Preprint*, arXiv:2410.23218.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and 1 others. 2024b. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

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

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

Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James Landay, and Monica S Lam. 2021. Grounding open-domain instructions to automate web support tasks. *Preprint*, arXiv:2103.16057.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025a. Aguvis: Unified pure vision agents for autonomous gui interaction. *Preprint*, arXiv:2412.04454.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and

Caiming Xiong. 2025b. Aguvis: Unified pure vision agents for autonomous gui interaction. *Preprint*, arXiv:2412.04454.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. 2025a. Magma: A foundation model for multimodal ai agents. *Preprint*, arXiv:2502.13130.

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

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2023. Webshop: Towards scalable real-world web interaction with grounded language agents. *Preprint*, arXiv:2207.01206.

Faraz YazdaniBanafsheDaragh and Sam Malek. 2022. Deep gui: black-box gui input generation with deep learning. In *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*, ASE '21, page 905–916. IEEE Press.

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

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2025. Large language model-brained gui agents: A survey. *Preprint*, arXiv:2411.18279.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Ufo: A ui-focused agent for windows os interaction. *Preprint*, arXiv:2402.07939.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023. Appagent: Multimodal agents as smartphone users. *Preprint*, arXiv:2312.13771.

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

A Training and Inference Details

A.1 Source Training Data

Following GUI-Actor (Wu et al., 2025), we compile our training dataset from several publicly available, high-quality GUI datasets, with summary statistics provided in Tab. 5. To ensure fair evaluation, we also exclude any samples from Wave-UI that overlap with the test sets of downstream tasks.

B Benchmarks

Our evaluation centers on six sophisticated benchmarks for GUI visual grounding:

ScreenSpot-v2 (Wu et al., 2024b) encompasses 1,272 carefully annotated instructions, each paired with corresponding target elements across diverse GUI environments, including mobile (Android and iOS), desktop (macOS and Windows), and web platforms. The dataset is designed to improve the quality and reliability of GUI visual grounding tasks, addressing key challenges such as eliminating ambiguities in natural language instructions and resolving annotation errors. By refining the alignment between textual descriptions and interface elements, ScreenSpot-v2 provides a robust and standardized benchmark for evaluating grounding models.

ScreenSpot-Pro (Li et al., 2025), meanwhile, focuses on more demanding scenarios, especially those involving high-resolution professional applications. It contains 1,581 tasks annotated by domain experts across 23 specialized software applications, spanning three operating systems. This benchmark significantly broadens the scope of GUI visual grounding by introducing interfaces with industrial software and multi-window layouts, creating a larger domain gap compared to most pretraining data. With its increased complexity and domain diversity, ScreenSpot-Pro is an invaluable resource for assessing the generalization ability of models in realistic and challenging GUI environments.

OSWorld-G is the grounding-specific subset derived from the OSWorld benchmark (Xie et al., 2025a), a unified evaluation environment for multimodal agents on Ubuntu. Unlike static datasets, OSWorld-G consists of screenshots captured from a fully functional, interactive operating system. It evaluates the model's ability to localize actionable elements within dynamic and complex real-world desktop workflows, serving as a direct proxy for an agent's practical utility in autonomous computer control tasks.

UI-Vision (Element Grounding) (Nayak et al., 2025) is designed to rigorously test the semantic understanding of user interface elements. While standard grounding tasks often rely on text matching (OCR), UI-Vision focuses on functional icons and visual symbols (e.g., identifying a "magnifying glass" as "search" or a "floppy disk" as "save") that lack explicit textual labels. Performance on this benchmark reflects the model's capacity for visual reasoning and its ability to interpret the functional affordances of GUI components.

UI-I2E (Image-to-Element) (Liu et al., 2025a) evaluates the capability to parse the hierarchical structure of a screen. The task requires the model to map raw pixel inputs to structured representations, effectively "reading" the underlying layout or accessibility tree of the interface. High accuracy on UI-I2E indicates that the model possesses a deep understanding of UI composition and element spatial relationships, rather than merely memorizing surface-level patterns.

MMBench-GUI L2 (Liu et al., 2024) is the GUI-specific subset (L-2 category) of the massive MMBench suite. Adopting a robust CircularEval strategy with multiple-choice questions, it assesses fine-grained perception and reasoning abilities within graphical interfaces. This benchmark serves as a standardized indicator of the model's general-purpose multimodal intelligence in the GUI domain, complementing the pure localization metrics of ScreenSpot.

C Detailed Experimental Results on ScreenSpot-v2 and ScreenSpot-Pro

We provide extended experimental results, including fine-grained performance breakdowns and comparisons against a broader set of baselines. Detailed statistics are presented in Tab. 6 and Tab. 7.

D Qualitative Analysis and Case Studies

D.1 Success Cases

Fig. 4 demonstrate several representative success cases where our V2P-7B model achieves accurate GUI element localization. Through these successful examples, we observe that the model exhibits high confidence in precisely highlighting target regions, with attention distributions that closely align with the actual shapes of UI elements. The attention maps show sharp, well-defined boundaries that accurately correspond to button edges, text field borders, and icon contours. This demonstrates the

Dataset	# of Elements	# of Screenshots	Platform
Uground Web–Hybrid (Gou et al., 2025a)	8M	775K	Web
GUI-Env (Chen et al., 2025)	262K	70K	Web
GUI-Act (Chen et al., 2025)	42K	13K	Web
AndroidControl (Li et al., 2024)	47K	47K	Android
AMEX (Chai et al., 2025)	1.2M	100K	Android
Wave-UI	50K	7K	Hybrid
Total	9.6M	1M	–

Table 5: Overview of training datasets used for GUI-Actor.

Model	ScreenSpot-v2 Accuracy (%)						
	Mobile-Text	Mobile-Icon	Desktop-Text	Desktop-Icon	Web-Text	Web-Icon	Avg.
Proprietary Models							
Operator	47.3	41.5	90.2	80.3	92.8	84.3	70.5
GPT-4o + OmniParser-v2	95.5	74.6	92.3	60.9	88.0	59.6	80.7
General Open-source Models							
Qwen2.5-VL-3B	93.4	73.5	88.1	58.6	88.0	71.4	80.9
Qwen2.5-VL-7B	97.6	87.2	90.2	74.2	93.2	81.3	88.8
GUI-specific Models (SFT)							
SeeClick-9.6B	78.4	50.7	70.1	29.3	55.2	32.5	55.1
Magma-8B	62.8	53.4	80.0	57.9	67.5	47.3	61.5
OS-Atlas-4B	87.2	59.7	72.7	46.4	85.9	63.1	71.9
UI-TARS-2B	95.2	79.1	90.7	68.6	87.2	78.3	84.7
OS-Atlas-7B	95.2	75.8	90.7	63.6	90.6	77.3	84.1
Aguvis-7B	95.5	77.3	95.4	77.9	91.0	72.4	86.0
UGround-V1-7B	95.0	83.3	95.0	77.8	92.1	77.2	87.6
UI-TARS-72B	94.8	86.3	91.2	87.9	91.5	87.7	90.3
GUI-Actor-3B	97.6	83.4	96.9	83.6	94.0	85.7	91.0
UI-TARS-7B	96.9	89.1	95.4	85.0	93.6	85.2	91.6
GUI-Actor-7B	97.6	88.2	96.9	85.7	93.2	86.7	92.1
GUI-specific Models (RL)							
SE-GUI-7B	-	-	-	-	-	-	90.3
LPO-8B	-	-	-	-	-	-	90.5
Ours							
V2P-7B	**98.1**	88.0	96.1	**89.7**	**95.4**	84.4	**92.4**

Table 6: Comparison of Model Performance Across Task Categories in ScreenSpot-v2. Bold text highlights the best results, while "–" represents missing values not reported in the original papers.

model's robust understanding of visual-semantic correspondence between natural language instructions and GUI components, effectively bridging the gap between textual descriptions and visual interface elements.

D.2 Failure Cases and Error Analysis

Our analysis of failure cases reveals several interesting patterns and limitations, as illustrated in Fig. 5. In some instances, we observe that the model encounters difficulties when multiple UI elements share semantic similarities. The model often exhibits high confidence while incorrectly selecting semantically related but functionally different elements or misidentifying similar icons with different purposes (Fig. 5a).

Additionally, we identify cases where the model's attention distribution becomes highly dispersed across the interface, which we interpret as an indicator of *low confidence* (Fig. 5b). This scattered attention pattern typically occurs in scenarios with numerous distracting elements or cluttered interfaces, suggesting that the model's decision-making process becomes uncertain when faced with complex visual layouts.

Furthermore, we observe failure modes where the model's attention concentrates entirely on regions completely unrelated to the target element (Fig. 5c). These cases often involve ambiguous natural language descriptions or interfaces with un-

Model	ScreenSpot-Pro Accuracy (%)														
	CAD		Dev		Creative		Scientific		Office		OS		Avg.		
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	**Avg.**
Proprietary Models															
GPT-4o	2.0	0.0	1.3	0.0	1.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	1.3	0.0	0.8
Claude Computer Use	14.5	3.7	22.0	3.9	25.9	3.4	33.9	15.8	30.1	16.3	11.0	4.5	23.4	7.1	17.1
General Open-source Models															
Qwen2.5-VL-3B	9.1	7.3	22.1	1.4	26.8	2.1	38.2	7.3	33.9	15.1	10.3	1.1	23.6	3.8	16.1
Qwen2.5-VL-7B	16.8	1.6	46.8	4.1	35.9	7.7	49.3	7.3	52.5	20.8	37.4	6.7	38.9	7.1	26.8
GUI-specific Models (SFT)															
SeeClick-9.6B	2.5	0.0	0.6	0.0	1.0	0.0	3.5	0.0	1.1	0.0	2.8	0.0	1.8	0.0	1.1
FOCUS-2B	7.6	3.1	22.8	1.7	23.7	1.7	25.0	7.1	23.2	7.7	17.8	2.5	19.8	3.9	13.3
CogAgent-18B	7.1	3.1	14.9	0.7	9.6	0.0	22.2	1.8	13.0	0.0	5.6	0.0	12.0	0.8	7.7
Aria-UI	7.6	1.6	16.2	0.0	23.7	2.1	27.1	6.4	20.3	1.9	4.7	0.0	17.1	2.0	11.3
OS-Atlas-7B	12.2	4.7	33.1	1.4	28.8	2.8	37.5	7.3	33.9	5.7	27.1	4.5	28.1	4.0	18.9
ShowUI-2B	2.5	0.0	16.9	1.4	9.1	0.0	13.2	7.3	15.3	7.5	10.3	2.2	10.8	2.6	7.7
UGround-7B	14.2	1.6	26.6	2.1	27.3	2.8	31.9	2.7	31.6	11.3	17.8	0.0	25.0	2.8	16.5
UGround-V1-7B	15.8	1.2	51.9	2.8	47.5	9.7	57.6	14.5	60.5	13.2	38.3	7.9	45.2	8.1	31.1
UI-TARS-2B	17.8	4.7	47.4	4.1	42.9	6.3	56.9	17.3	50.3	17.0	21.5	5.6	39.6	8.4	27.7
UI-TARS-7B	20.8	9.4	58.4	12.4	50.0	9.1	63.9	31.8	63.3	20.8	30.8	16.9	47.8	16.2	35.7
UI-TARS-72B	18.8	12.5	62.9	17.2	57.1	15.4	64.6	20.9	63.3	26.4	42.1	15.7	50.9	17.6	38.1
JEDI-3B	27.4	9.4	61.0	13.8	53.5	8.4	54.2	18.2	64.4	32.1	38.3	9.0	49.8	13.7	36.1
JEDI-7B	38.0	14.1	42.9	11.0	50.0	11.9	72.9	25.5	75.1	47.2	33.6	16.9	52.6	18.2	39.5
GUI-Actor-7B	–	–	–	–	–	–	–	–	–	–	–	–	–	–	44.6
GUI-specific Models (RL)															
UI-R1-3B	11.2	6.3	22.7	4.1	27.3	3.5	42.4	11.8	32.2	11.3	13.1	4.5	24.9	6.4	17.8
UI-R1-E-3B	37.1	12.5	46.1	6.9	41.9	4.2	56.9	21.8	65.0	26.4	32.7	10.1	–	–	33.5
GUI-R1-3B	26.4	7.8	33.8	4.8	40.9	5.6	61.8	17.3	53.6	17.0	28.1	5.6	–	–	–
GUI-R1-7B	23.9	6.3	49.4	4.8	38.9	8.4	55.6	11.8	58.7	26.4	42.1	16.9	–	–	–
InfiGUI-R1-3B	33.0	14.1	51.3	12.4	44.9	7.0	58.3	20.0	65.5	28.3	43.9	12.4	49.1	14.1	35.7
GUI-G1-3B	39.6	9.4	50.7	10.3	36.6	11.9	61.8	30.0	67.2	32.1	23.5	10.6	49.5	16.8	37.1
SE-GUI-3B	38.1	12.5	55.8	7.6	47.0	4.9	61.8	16.4	59.9	24.5	40.2	12.4	50.4	11.8	35.9
SE-GUI-7B	51.3	42.2	68.2	19.3	57.6	9.1	75.0	28.2	78.5	43.4	49.5	25.8	63.5	21.0	47.3
GUI-G^2-7B	55.8	12.5	68.8	17.2	57.1	15.4	77.1	24.5	74.0	32.7	57.9	21.3	64.7	19.6	47.5
Ours															
V2P-7B	**58.38**	12.50	67.53	**24.83**	**62.63**	**16.08**	73.61	**33.64**	75.71	43.40	56.07	**32.58**	**65.81**	**25.83**	**52.50**

Table 7: Comparison of Model Performance Across Task Categories in ScreenSpot-Pro. Bold text highlights the best results, while "–" represents missing values not reported in the original papers. The baseline models utilize various backbones and parameter sizes, as indicated by their names (e.g., -7B, -18B).

conventional design patterns that deviate from the model's training distribution. Such failures highlight the need for enhanced user intent understanding and more comprehensive UI context comprehension capabilities.

D.3 Multi-step Interaction Scenarios

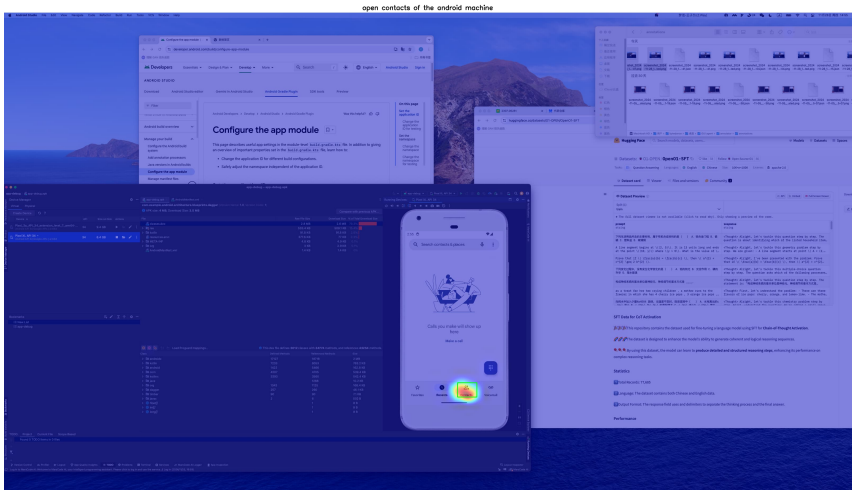
To visualize the model's capability in maintaining context across sequential operations, we present case studies of multi-step workflows from the AndroidControl (Li et al., 2024) dataset. Fig. 6a and Fig. 7 showcases the model's performance across sequential GUI operations.

The results demonstrate that our model maintains consistent accuracy throughout extended interaction sequences, successfully completing multi-step tasks that require contextual understanding and state awareness.

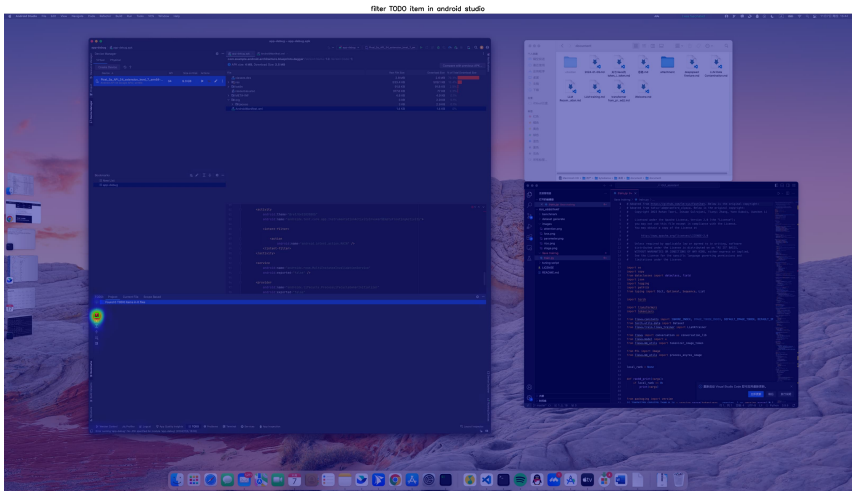
D.4 Multi-target Localization Capabilities

We investigated the model's ability to simultaneously localize multiple targets within a single interface, which holds significant value for batch operations and improving inference efficiency. Fig. 6b presents our experimental setup using a calculator interface, where we tasked the model with simultaneously localizing the elements "1", "0", and "00".

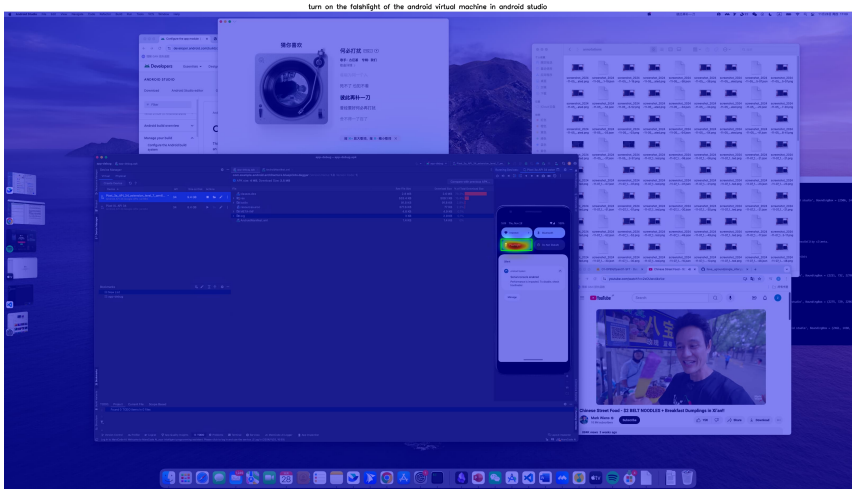
The results reveal that the model successfully generates attention distributions for all three target elements simultaneously, with appropriately differentiated confidence levels. Notably, the element "1" receives the highest attention intensity, followed by "0" and "00" respectively, which aligns with the natural priority of these elements. This multi-target capability demonstrates the model's sophisticated attention mechanism and its potential for complex GUI analysis tasks requiring simultaneous element identification, as well as its genuine understanding capability of user queries.



(a) Success Case 1

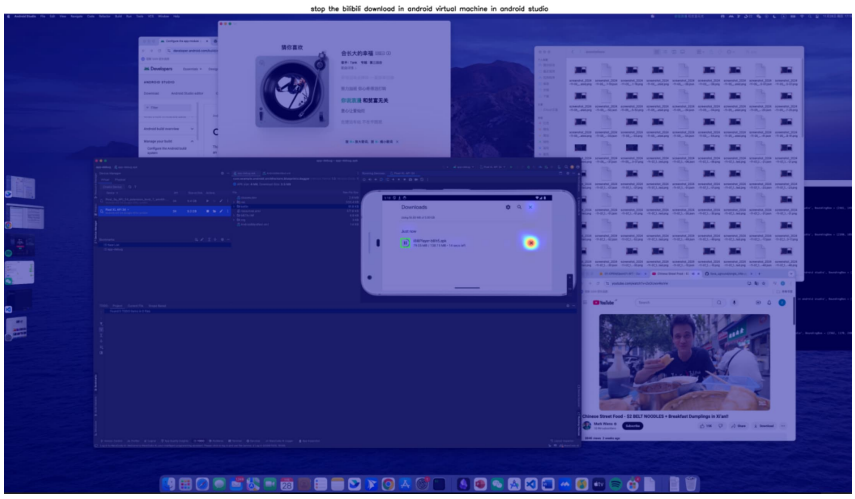


(b) Success Case 2

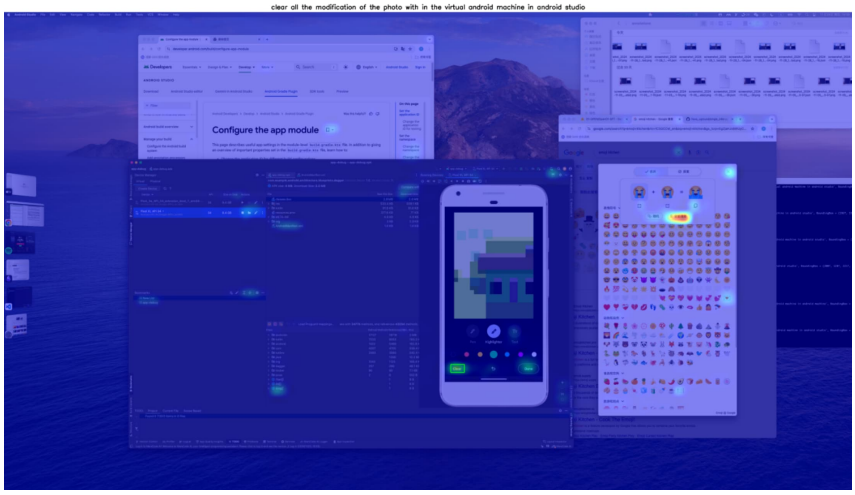


(c) Success Case 3

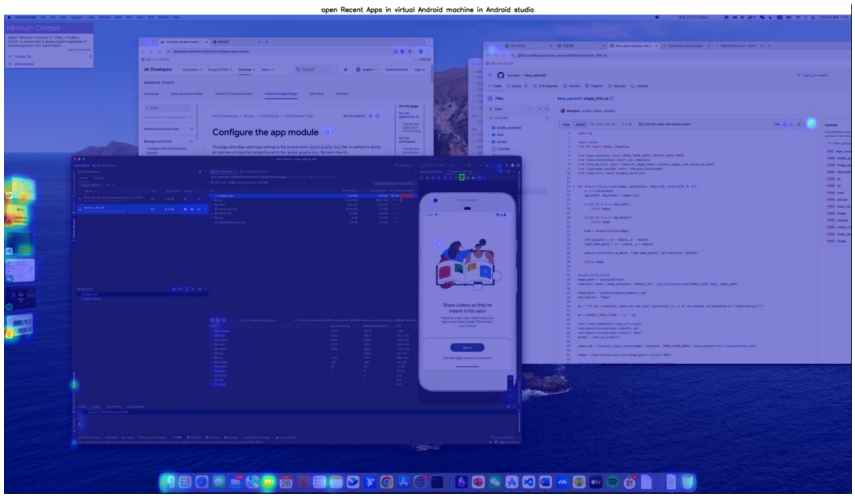
Figure 4: Representative success cases of GUI element localization.



(a) Failure Case 1

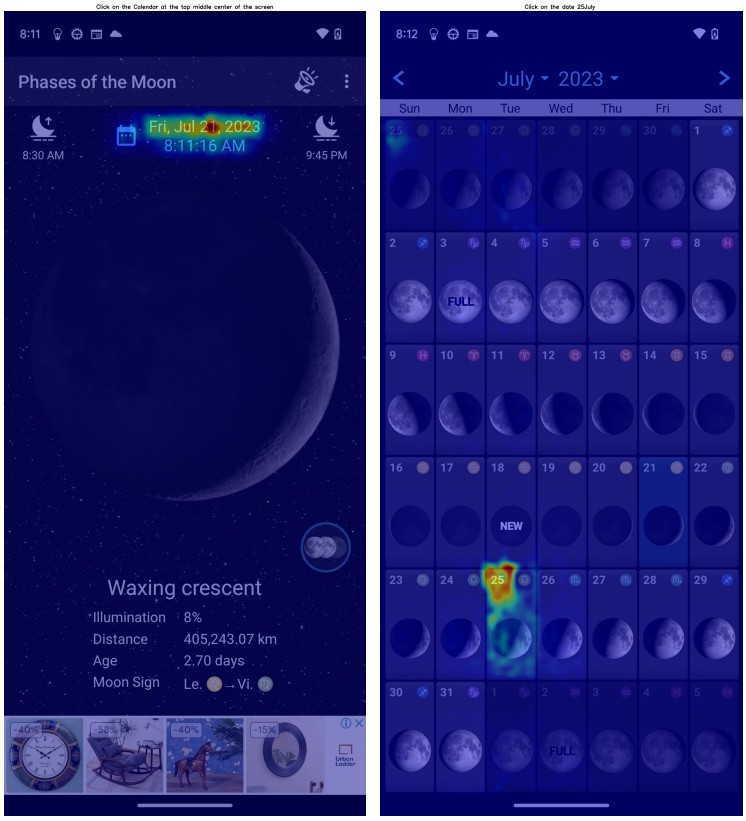


(b) Failure Case 2

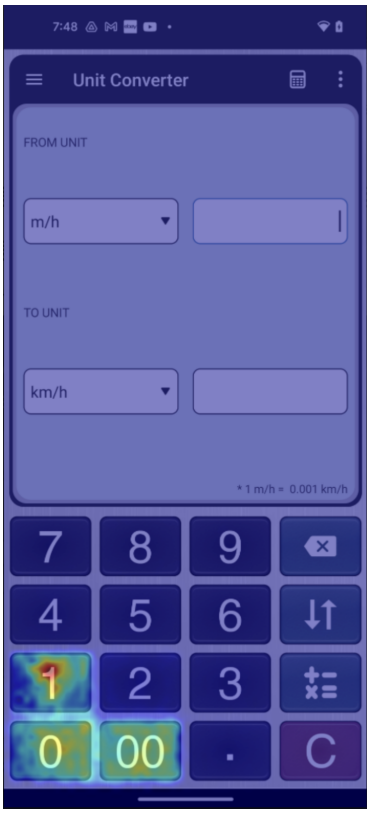


(c) Failure Case 3

Figure 5: Representative failure cases of GUI element localization.

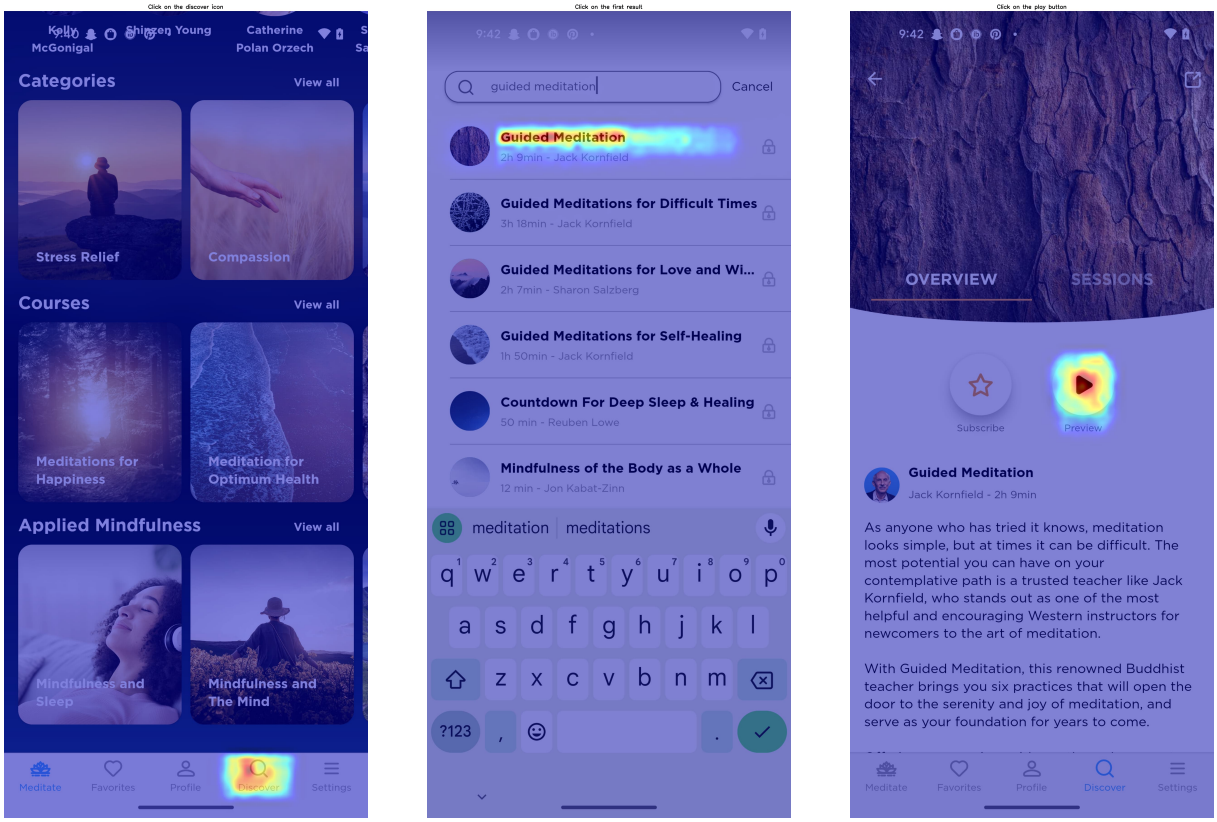


(a) Multi step grounding case 1: "Open Phase of the moon App, select the date 25 July on the calendar and view the moon phase for that date." Step 1 (left) and Step 2 (right).



(b) Multi-target grounding case.

Figure 6: Multi-step grounding case and multi-target grounding case.



(a) Step 1: Click on the discover icon. (b) Step 2: Click on the first result. (c) Step 3: Click on the play button.

Figure 7: Multi step grounding case 2: "Open the Mindfulness app, I would like to have a personalized guided meditation to help me be productive throughout the day."