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

Anonymous ACL submission

Abstract

Precise GUI element localization is fundamental for autonomous agents interacting with graphical interfaces. While recent attention-based methods show promise, they suffer from systematic failures: attention drifts to visually similar background UI elements, and uniform bounding box supervision fails to distinguish the element's actionable centers from peripheral regions. We propose Valley-to-Peak (V2P), a training framework that reshapes attention maps by simultaneously suppressing background regions and concentrating probability mass at target centers. V2P combines Suppression Attention, which penalizes non-target areas, with Gaussian Peak Modeling, which concentrates within-target attention at element centers using size-adaptive Gaussian distributions. Evaluation across six diverse benchmarks demonstrates substantial improvements, especially on challenging high-resolution and out-of-distribution scenarios. Attribution analysis confirms that 86.2% of performance gains directly address the two identified failure modes. Fine-grained metrics show a 28.6% improvement in center-region hit rate. Additionally, V2P achieves 9.3× inference speedup over autoregressive baselines, validating its practical utility for real-time GUI agent deployment.

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, outputting bounding boxes or (x, y) coordinates (Zhang et al., 2025; Qin et al., 2025). However, this suffers from weak spatial–semantic alignment (Wu et al., 2025), treating coordinates as ordinary tokens without inherent spatial meaning, and ignores that clicks are valid anywhere within a target. Recent work leverages attention maps (Wu et al., 2025): instead of predicting coordinates, it extracts cross-modal attention weights between special tokens and image patches, selecting the most attended patch. This offers dense spatial supervision and naturally accommodates multiple valid click regions.

Despite these advances, attention-based grounding exhibits systematic failures. We identify a fundamental problem: **learned attention distributions poorly align with valid interaction regions**. An ideal attention map should peak sharply at the target center and approach zero elsewhere. As Fig. 1 shows, existing methods produce flat, dispersed attention maps that fail in two ways: (1) **Background Distraction**: Existing loss functions reward attention on target patches but impose no explicit penalty on background regions. As a result, visually or semantically similar non-target elements accumulate substantial attention mass. Under softmax normalization, this "leaked" mass directly dilutes the attention on target elements, causing the predicted click to drift toward irrelevant regions; (2) **Centre-Edge Confusion**: Because supervision treats all patches within a bounding box as equally valid, the model has no incentive to distinguish an element's actionable center from its periphery. This produces uniform attention across the target, leading to imprecise clicks, particularly for small elements, where boundary attention can cause the prediction to fall outside the intended target entirely.

Both issues stem from the same root cause: the gap between training supervision and the true spa-

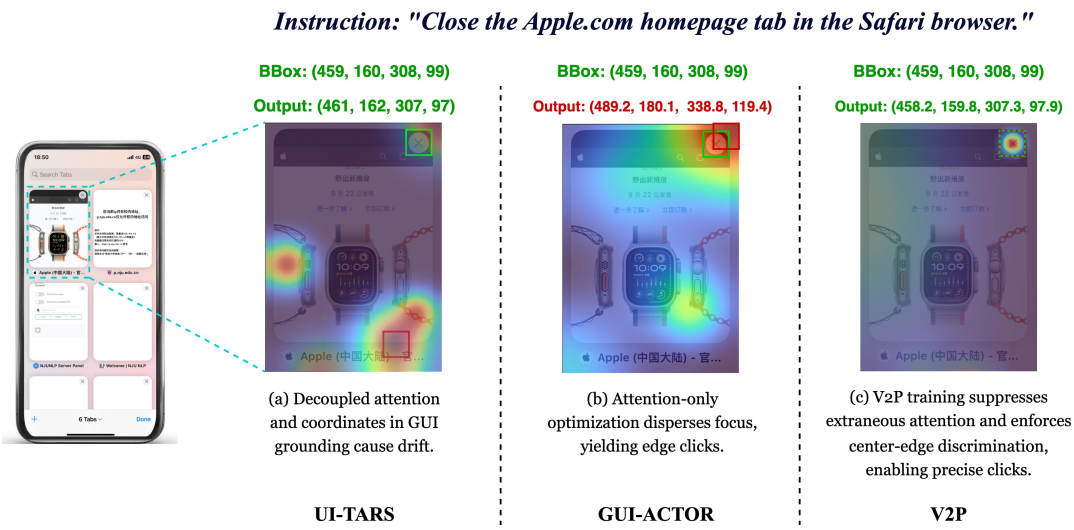


Figure 1: Comparison of different strategies in the GUI grounding task. The green box marks the ground-truth bounding box, and the red box highlights the region where the model places the highest attention given the instruction and screenshot. The overlaid heatmap is colour-coded from cool (blue) to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

tial distribution of interaction likelihood. We address this with **Valley-to-Peak (V2P)**, which reshapes the attention map from both ends simultaneously by suppressing where attention should be low, and sharpening where it should be high:

First, **Suppression Attention (SA)** applies an inverse-attention penalty to drive attention toward zero in non-target regions, carving out low-attention "valleys" that isolate the target from distractors. However, concentrating attention within the target region alone is insufficient: without finer spatial guidance, the model still distributes attention uniformly across the bounding box, failing to distinguish the actionable center from peripheral edges. Therefore, **Gaussian Peak Modeling (GPM)** provides a size-adaptive 2D Gaussian prior that peaks at the element's center and decays toward the edges, shaping the within-target distribution to reflect true interaction likelihood. Together, SA and GPM push the attention map toward its ideal shape: near-zero outside the target (valleys) and sharply peaked at the center (peak), constituting the Valley-to-Peak training paradigm.

Our contributions are as follows:

1. Through systematic analysis of attention maps in existing GUI grounding models, we identify the attention misalignment problem, which manifests as Background Distraction and Centre-Edge Confusion, both rooted in the mismatch between uniform bounding-box supervision and actual interaction-likelihood distributions.

2. We propose the V2P framework that jointly addresses both failure modes: SA drives background attention toward zero, while GPM shapes the target-region distribution to reflect spatial interaction priors. The two components complement each other as coarse (region-level suppression) and fine (within-target shaping) attention control.

3. Extensive experiments across six diverse benchmarks demonstrate that V2P consistently outperforms strong baselines under controlled settings, with substantial gains on challenging scenarios. Attribution analysis reveals that 86.2% of performance gains directly address the two identified failure modes. Fine-grained spatial metrics confirm V2P's superior center alignment and effectiveness on small targets. Moreover, V2P achieves 9.3× inference speedup, demonstrating strong practical utility for real-time GUI agent deployment.

2 Related Work

2.1 GUI-Agents

GUI agents have evolved from rule-based test tools (Wetzlmaier et al., 2016; Memon et al., 2001; Steven et al., 2000) to multimodal LLM-driven systems. Early learning-based methods (Li et al., 2020; YazdaniBanafsheDaragh and Malek, 2022) learned action policies from screenshots,

while widget detectors (White et al., 2019) improved element recognition. Natural-language interfaces (Mazumder and Riva, 2021; Xu et al., 2021) and RL environments (Shi et al., 2017; Yao et al., 2023) further advanced web-scale interaction. Recent LLM-powered agents unify 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, Operator) further attest to practical traction.

2.2 GUI Grounding

Prevalent approaches frame GUI grounding as coordinate generation (Zhang et al., 2025). Models such as UI-TARS (Qin et al., 2025) and CogAgent (Hong et al., 2024) train VLMs to autoregressively produce textual coordinates, but treating spatial values as ordinary tokens limits fine-grained visual alignment. Recent methods instead leverage cross-modal attention maps (Wu et al., 2025), deriving predictions from the highest-attended image patch. While more robust, this paradigm suffers from attention leaking into irrelevant regions or spreading uniformly across the target. Our work directly refines the attention map quality itself.

V2P draws inspiration from two areas. To suppress background noise and create attention "valleys," we adopt attention suppression techniques (Li et al., 2018). To form a sharp "peak" at the target center, we build on Fitts' Law from HCI (MacKenzie, 1992) and Gaussian heatmap regression from localization tasks (Fitts, 1954). To our knowledge, V2P is the first to synergistically combine background suppression with center-focused peak modeling for GUI grounding.

3 Method

We introduce Valley-to-Peak (V2P), a training framework that reshapes attention maps by simultaneously suppressing background regions and concentrating probability mass at centers of the targeted UI elements. It addresses two failure modes through complementary mechanisms: (1) **Suppression Attention (SA):** Applies an inverse-attention penalty to drive attention toward zero in non-target regions, carving low-attention valleys that isolate targets from distractors (addresses Background Distraction), and (2) **Gaussian Peak Modeling (GPM):** Replaces the uniform supervision within target bounding boxes with size-adaptive 2D Gaussian distributions, creating sharp attention peaks at actionable centers (addresses Centre-Edge Confusion).

By jointly optimizing these objectives, V2P produces spatially-aware attention maps that overcome the limitations of uniform supervision. Below, we outline the architecture (Sec. 3.1), then detail SA (Sec. 3.2) and GPM (Sec. 3.3).

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:

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

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

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

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

Attention scores are then computed as:

$$\alpha_i = \frac{z^\top z_i}{\sqrt{d}}, \quad a_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{M} \exp(\alpha_j)} \quad (4)$$

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

As identified in our analysis, attention-based methods suffer from *Background Distraction*, where attention drifts to visually similar non-target elements. To address this, we propose Suppression Attention (SA), which explicitly penalizes attention allocated to non-target regions, driving attention

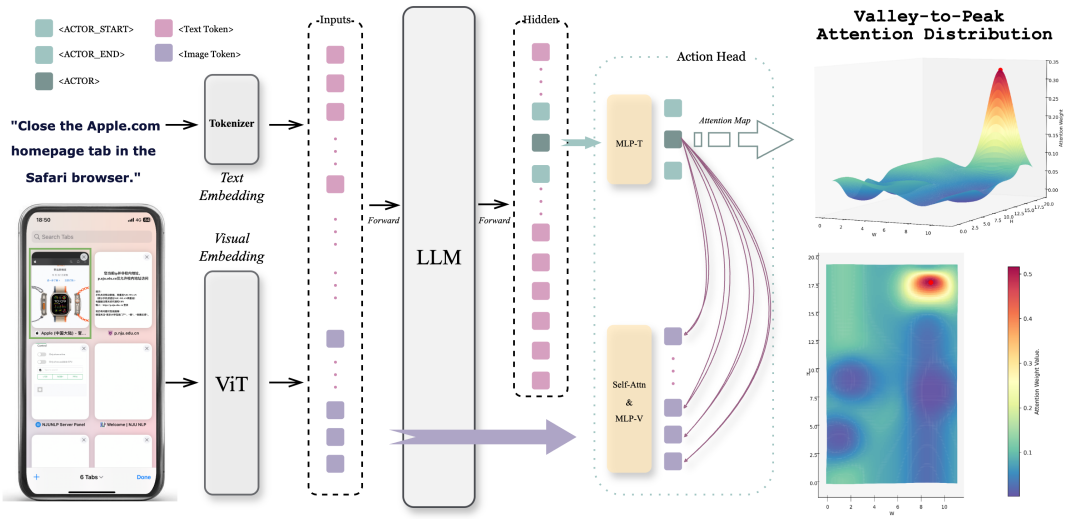


Figure 2: **Valley-to-Peak Training Framework.** V2P reshapes attention maps through two complementary mechanisms: Suppression Attention (SA) penalizes non-target regions to create valleys, while Gaussian Peak Modeling (GPM) uses size-adaptive Gaussian distributions to create sharp peaks at element centers. This dual approach enables precise localization in cluttered interfaces.

toward zero outside the target and isolating it from distractors.

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 Gaussian Peak Modeling

While SA suppresses background regions, uniform supervision within target bounding boxes leads to *Centre-Edge Confusion*, where the model cannot distinguish actionable centers from peripheries, resulting in imprecise clicks especially for small or overlapping elements.

To address this, we propose Gaussian Peak Modeling (GPM), which replaces uniform labels with size-adaptive 2D Gaussian distributions. Inspired by Fitts' Law (MacKenzie, 1992; Fitts, 1954), which shows click probability peaks at element centers and decays toward edges, GPM provides a soft spatial prior that reflects interaction likelihood distributions.

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:

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

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.

GPM establishes a center-biased, size-aware attention prior that reflects human interaction patterns. By providing smooth spatial gradients from edges to centers, it enhances localization precision and resolves Centre-Edge Confusion in complex 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 $\mathcal{L}_{\text{Sup_Attn}}$ drives attention toward zero in non-target regions (SA, Sec. 3.2), and $\mathcal{L}_{\text{Action_Attn}}$ aligns the predicted attention with Gaussian-shaped target distributions (GPM, Sec. 3.3).

Minimizing this combined loss realizes the Valley-to-Peak paradigm: SA creates valleys by suppressing background regions, while GPM creates peaks by concentrating probability mass at target centers. Together, they reshape attention from flat, dispersed distributions into sharp, centered peaks, addressing both Background Distraction and Centre-Edge Confusion for precise localization in complex GUI environments.

4 Experiment

4.1 Experimental Setup

We utilize Qwen2.5-VL-Instruct (both 3B and 7B) (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. The Gaussian factor $\sigma = 1.0$ is used consistently across all experiments to ensure reproducibility and fair comparison. Comprehensive details are provided in App. A and App. E.1.

We evaluate on a comprehensive suite of six benchmarks. Our primary evaluation focuses on *ScreenSpot-v2* (Wu et al., 2024) 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., 2024), *UI-Vision* (Element Grounding) (Nayak et al., 2025), *UI-I2E* (Liu et al., 2025), and *MMBench-GUI L2* (Liu et al., 2024).

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	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**

4.2 Main Results

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

Superior Performance on ScreenSpot Benchmarks. As our primary evaluation targets, 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 out-of-distribution applications, V2P-7B attains 52.5%, substantially outperforming the strong baseline GUI-Actor-7B (44.6%) by 7.9 percentage points and UI-TARS-72B (38.1%) by 14.4 points. This substantial margin validates that V2P's attention calibration mechanism is particularly effective in handling 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 evaluation to four benchmarks featuring interaction traces and functional reasoning requirements: *OSWorld-G* (Xie et al., 2024), *UI-Vision* (Element Grounding) (Nayak et al., 2025), *UI-I2E* (Liu et al., 2025), 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* (28.8%), *UI-I2E* (75.6%), and *MMBench-GUI L2* (79.9%). This consistent superiority highlights the model's exceptional functional reasoning and semantic understanding capabilities. Furthermore, on *OSWorld-G*, V2P matches the specialist JEDI-7B (Xie et al., 2025) (52.5%) despite using only approximately 50k PC samples versus JEDI's millions, demonstrating superior data efficiency and strong 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. Remarkably, on some challenging benchmarks such as ScreenSpot-Pro (48.5%), it even surpasses significantly larger-scale models, demonstrating that V2P's gains stem from algorithmic innovation rather than model capacity. This result underscores the pure algorithmic superiority of the V2P framework and its consistent effectiveness across varying model scales.

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

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

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.

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%

4.3 Ablation and Analysis

4.3.1 Component Ablation Study

To validate the necessity of our proposed components, we conduct a systematic ablation study on *ScreenSpot-Pro* (Tab. 2). Removing Gaussian Peak Modeling leads to a significant performance drop of 5.0 percentage points, confirming its critical role in precise center localization. Further removing Suppression Attention results in an additional loss of 3.2 points, bringing the model down to the baseline level. These results verify that both modules are indispensable and work synergistically within the V2P framework.

4.3.2 Attribution of Performance Gains

To investigate the underlying mechanisms behind V2P's superior performance, we conduct a quantitative attribution analysis on 182 samples where V2P-7B successfully corrects failures from 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 35.7% are attributed to resolving *Center-Edge Confusion*. Together, these two targeted improvements account for 86.2% of all gains, providing strong empirical evidence that V2P's dual-loss mechanism functions precisely as designed to address the identified failure modes.

4.3.3 Fine-Grained Spatial Precision

To verify that V2P effectively resolves Centre-Edge Confusion beyond binary accuracy metrics, we conduct pixel-level analysis on *ScreenSpot-Pro*. We introduce the **Centre-50% Hit Rate**, which measures the percentage of predicted clicks falling within the central 50% area of the target bounding box, representing the most robust actionable zone for user interactions.

Table 4: **Fine-Grained Spatial Precision on ScreenSpot-Pro.** V2P achieves substantially higher centre-region hit rate, confirming its effectiveness in resolving Centre-Edge Confusion.

Metric	GUI-Actor	V2P
Centre-50% Hit Rate	27.64%	**35.55%**

As shown in Tab. 4, V2P-7B achieves a Centre-50% Hit Rate of 35.55%, compared to GUI-Actor-7B's 27.64%, representing an obvious improvement. This provides direct empirical evidence that V2P fundamentally shifts attention distribution from edges to actionable centres, confirming its effectiveness in addressing Centre-Edge Confusion.

4.3.4 Performance Leap on Tiny Targets

We categorize UI elements by area relative to patch size n (14×14 pixels): Small ($n \leq A < 4n$), Medium ($4n \leq A < 9n$), and Large ($A \geq 9n$). As shown in Tab. 5, V2P-7B outperforms GUI-Actor-7B (Wu et al., 2025) by 10.0% on small elements in ScreenSpot-v2 and 6.3% on ScreenSpot-Pro, demonstrating superior fine-grained localization.

The size-stratified analysis reveals why performance gaps vary across benchmarks. *ScreenSpot-v2* is dominated by large elements whose generous boundaries tolerate imprecise localization, masking the baseline's deficiencies. *ScreenSpot-Pro* contains dense small elements requiring precise positioning, where V2P's advantage is fully realized.

V2P shows slight decline on large elements in *ScreenSpot-v2*, as the baseline's diffuse attention occasionally covers oversized targets despite imprecision, while V2P's sharp center-focused attention sacrifices marginal coverage. However, this trade-off is favorable, yielding dominant gains on small and medium categories where precision matters most.

Table 5: **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.

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%**

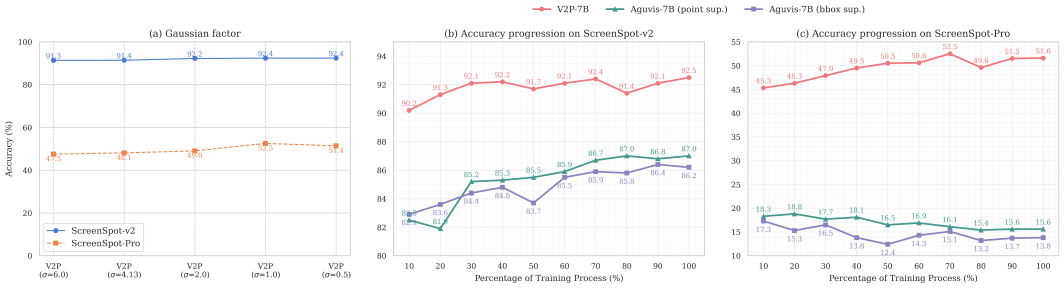


Figure 3: **Hyperparameter Sensitivity and Training Stability Analysis.** (a) Impact of Gaussian Factor σ on ScreenSpot-v2 and ScreenSpot-Pro. Optimal performance is achieved at $\sigma = 1.0$, balancing spatial precision and label smoothness. (b, c) Training curves comparing V2P-7B and Aguvis-7B. V2P demonstrates consistent improvement on both in-distribution and OOD benchmarks, while the baseline suffers from overfitting on OOD data after early training stages.

4.3.5 Inference Efficiency

Beyond accuracy, practical GUI agents require real-time responsiveness. V2P employs a single-pass pointer network that predicts a single $<$ ACTOR $>$ token to indicate the target location, whereas autoregressive methods like Qwen2.5-VL-7B require multiple forward passes to sequentially generate coordinate digits. As shown in Tab. 6 (experiment setup details in App. D), V2P-7B achieves 102.79ms per sample on *ScreenSpot-v2*, a 9.3× speedup over Qwen2.5-VL-7B's 957.49ms. On the high-resolution *ScreenSpot-Pro*, where image encoding dominates latency, V2P still maintains a 2.1× advantage. This efficiency gain, combined with superior accuracy, positions V2P as a practical solution for real-time GUI automation.

Table 6: **Inference Latency (ms).** V2P achieves substantial speedup over autoregressive baseline (Qwen2.5-VL-7B) on both ScreenSpot-v2 and ScreenSpot-Pro.

Dataset	Model	Latency (ms) ↓	Speedup ↑
ScreenSpot-v2	Qwen-7B	957.49	1.0×
	V2P-7B	**102.79**	**9.3×**
ScreenSpot-Pro	Qwen-7B	2608.49	1.0×
	V2P-7B	**1263.40**	**2.1×**

4.3.6 Hyperparameter Robustness

The Gaussian factor σ controls the spatial concentration of the attention prior. As shown in Fig. 3(a) and App. E, performance is stable across a range of values: on *ScreenSpot-v2*, accuracy varies only 1.1% between σ=0.5 and σ=6.0. We set σ=1.0 as a universal default, which yields top-tier results across all six benchmarks, spanning both easy and challenging scenarios, without per-dataset tuning, confirming that V2P is robust to this hyperparameter choice.

4.3.7 Training Stability and Generalization

We compare training dynamics of V2P-7B against Aguvis-7B (Xu et al., 2025) in Fig. 3(b,c) and App. E. V2P-7B shows consistently ascending accuracy on both in-distribution (*ScreenSpot-v2*) and OOD (*ScreenSpot-Pro*) benchmarks throughout training. In contrast, Aguvis-7B exhibits overfitting: its OOD performance declines after the 20% training milestone, suggesting that coordinate-based supervision fails to learn generalizable spatial reasoning. V2P's attention-based supervision provides spatially grounded inductive biases that transfer across diverse GUI environments.

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. V2P synergizes *Suppression Attention* to eliminate background noise and *Gaussian Peak Modeling* to construct size-adaptive peaks at actionable centers. This unified framework enables both precise localization and robust generalization across diverse GUI environments.

Experiments across diverse benchmarks demonstrate that V2P consistently outperforms strong baselines while achieving significant inference speedup over autoregressive alternatives. Our analyses validate the effectiveness of the dual-loss mechanism, showing substantial improvements in both overall accuracy and fine-grained spatial precision. These results suggest that attention calibration through explicit spatial priors is a promising direction for building precise and efficient perception modules for autonomous GUI agents.

Limitations

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

- **Dependence on Bounding-Box Annotations:** V2P's Gaussian Peak Modeling leverages bounding-box annotations to compute size-adaptive Gaussian priors during training. While this provides optimal supervision, the framework can gracefully degrade to point-click annotations by adopting a fixed-size Gaussian centered at the click point, at the cost of losing size-adaptive concentration. In practice, this limitation is minor, as bounding-box annotation tooling is mature and widely available for GUI datasets.

- **Static Single-Frame Input:** V2P operates on individual screenshots without leveraging temporal context from interaction sequences. In dynamic scenarios such as animations, transitions, or multi-step workflows, incorporating historical frames could further improve grounding robustness, which we leave to future work.

- **Scalability to Ultra-High Resolution:** Although V2P achieves substantial inference speedup over autoregressive baselines (Sec. 4.3.5), its self-attention module over visual patches scales quadratically with token count. For ultra-high-resolution inputs (e.g., 8K displays) where patches grow significantly, this quadratic cost may become a bottleneck, necessitating efficient attention mechanisms such as sliding-window or sparse attention.

Ethical Considerations

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.

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

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.

Xinyi Liu, Xiaoyi Zhang, Ziyun Zhang, and Yan Lu. 2025. 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.

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.

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.

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 1 others. 2024. 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. 2025. Scaling computer-use grounding via user interface decomposition and synthesis. *Preprint*, arXiv:2505.13227.

Tianbao Xie, Danyang Zhang, Junda Chen, Mulong Xie, Ziniu Hu, Shuai Wang, Zhoujun Cheng, Yiheng Xu, Jiangjie Chen, Chao Tang, Huazhe Xu, Luke Zettlemoyer, Yingjun Lyu, Yue Wang, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Preprint*, arXiv:2404.07972.

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

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.

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.

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. 7. 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., 2024) 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., 2024), 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., 2025) 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. 8 and Tab. 9.

D Inference Benchmark Details

All latency measurements were conducted on a single NVIDIA H200 GPU. To ensure a fair comparison, both the baseline Qwen2.5-VL-7B and V2P-7B were evaluated using an identical protocol: 20 samples for warm-up to stabilize GPU states, followed by inference on a fixed batch of 100 samples. We repeated the test 3 times and report the average end-to-end latency and throughput.

Table 7: Overview of training datasets used for GUI-Actor.

Dataset	# of Elements	# of Screenshots	Platform
Uground Web–Hybrid (Gou et al., 2025)	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 8: 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	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**

E Hyperparameter Analysis and Training Stability

E.1 Sensitivity to Gaussian Factor σ

The Gaussian factor σ controls the spatial spread of the Gaussian distribution in our peak modeling. To analyze its impact 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 exhibits sensitivity 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 performance decline.

We attribute this phenomenon to the spatial concentration of the attention mechanism. Larger σ values generate broader Gaussian distributions, which tend to dilute spatial focus and introduce background noise into attention maps. Conversely, smaller σ produces sharper Gaussian peaks, acting as tight spatial constraints that enable more precise localization. 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.

Importantly, all experiments reported in the main paper use $\sigma = 1.0$ consistently to ensure fair comparison and reproducibility. The hyperparameter was determined once during preliminary experi-

Table 9: 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).

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**

ments and held fixed across all benchmarks and ablation studies, demonstrating the robustness of our approach without extensive tuning.

E.2 Training Stability and Generalization

We evaluate the training stability of V2P-7B compared to Aguvis-7B (Xu et al., 2025), 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 throughout training.

In sharp contrast, Aguvis-7B exhibits a distinct overfitting pattern: while its performance improves on *ScreenSpot-v2*, it suffers from continuous performance decline on the OOD *ScreenSpot-Pro* after the 20% training milestone. This divergence suggests that coordinate-based supervision, while effective for in-distribution data, fails to capture generalizable spatial reasoning patterns.

V2P's superior generalization can be attributed to its attention-based supervision, which encourages the model to learn robust spatial priors (via Gaussian Peak Modeling) and background suppression strategies (via Suppression Attention). These mechanisms provide spatially grounded inductive biases that transfer across diverse GUI environments, effectively mitigating overfitting to specific coordinate patterns or visual styles present in the training distribution.

F Qualitative Analysis and Case Studies

F.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 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.

F.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 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 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 unconventional 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.

F.3 Multi-step Interaction Scenarios

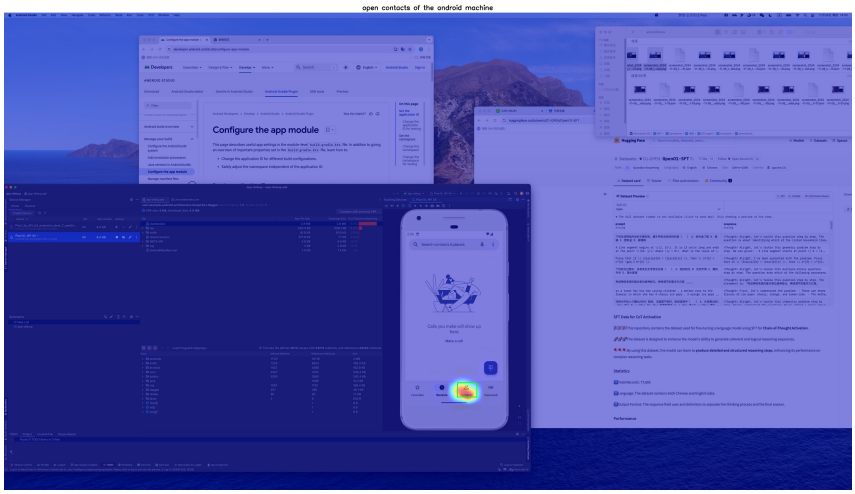
To visualize the model's capability in maintaining context across sequential operations, we present case studies of multi-step workflows from the AndroidControl (Li et al., 2024) dataset. Fig. 6a and Fig. 7 showcases the model's performance across sequential GUI operations.

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.

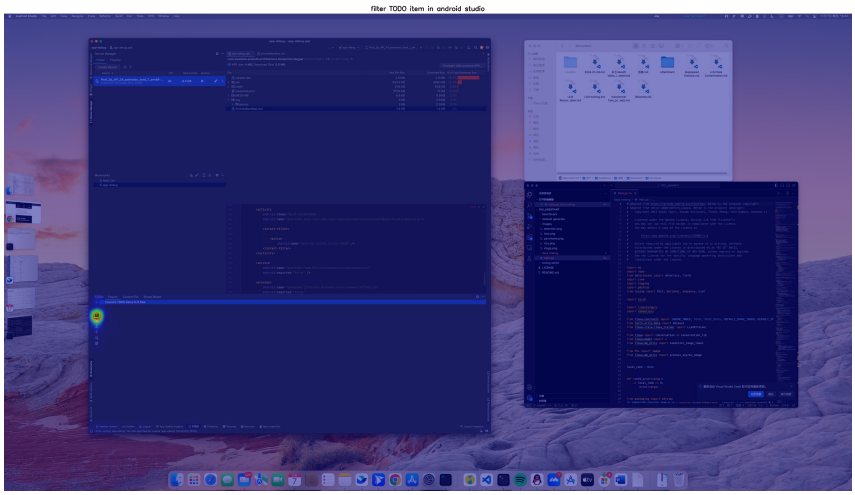
F.4 Multi-target Localization Capabilities

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".

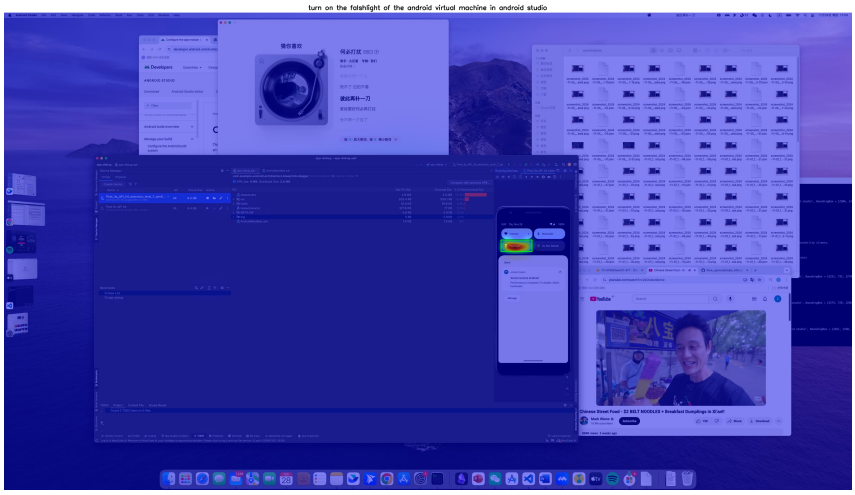
The results reveal that the model successfully generates attention distributions for all three target elements simultaneously, with appropriately differentiated confidence levels. Notably, the element "1" receives the highest attention intensity, followed by "0" and "00" respectively, which aligns with 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

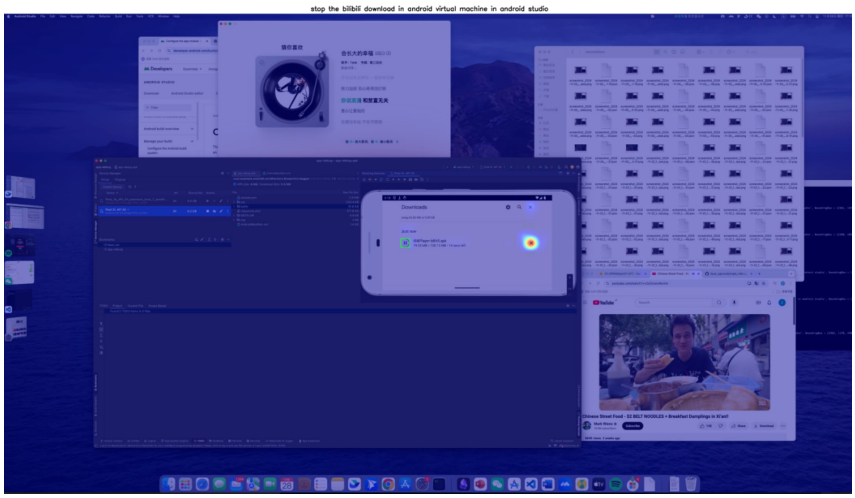


(b) Success Case 2

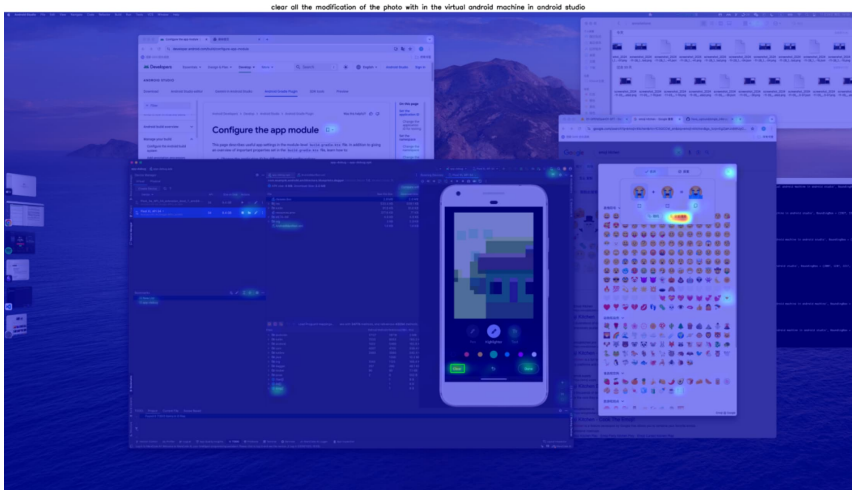


(c) Success Case 3

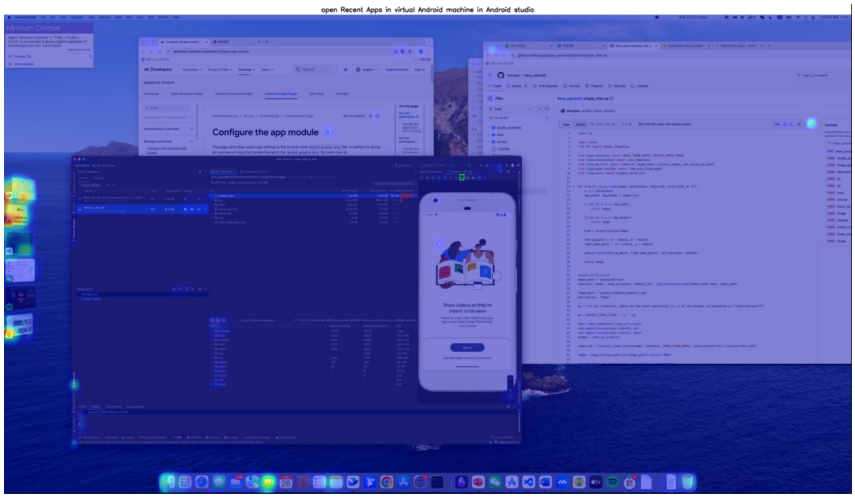
Figure 4: Representative success cases of GUI element localization.



(a) Failure Case 1

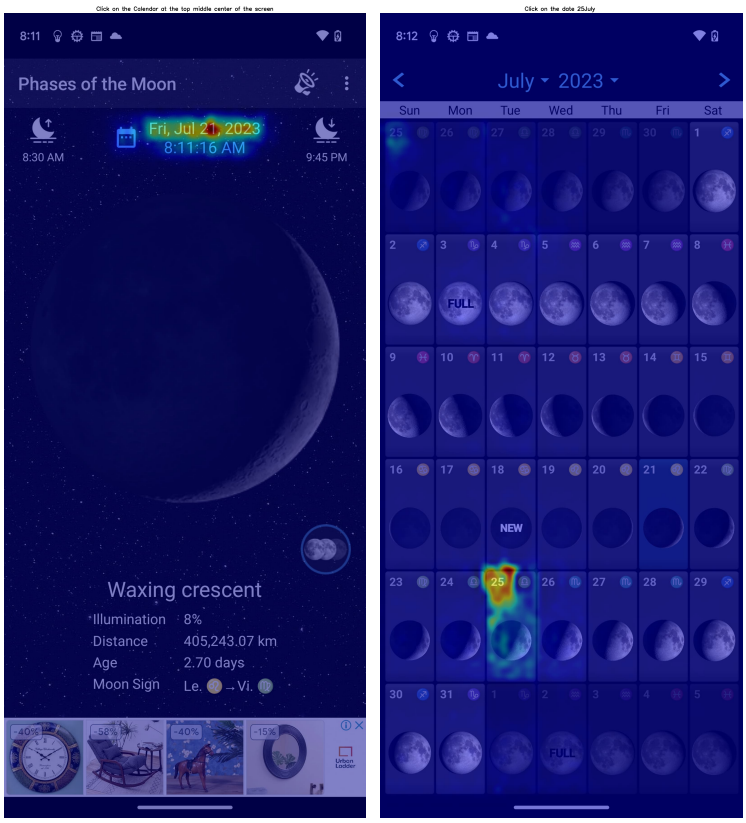


(b) Failure Case 2

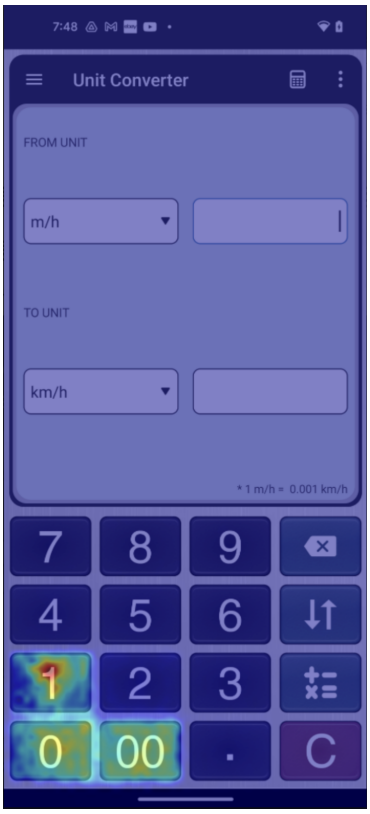


(c) Failure Case 3

Figure 5: Representative failure cases of GUI element localization.

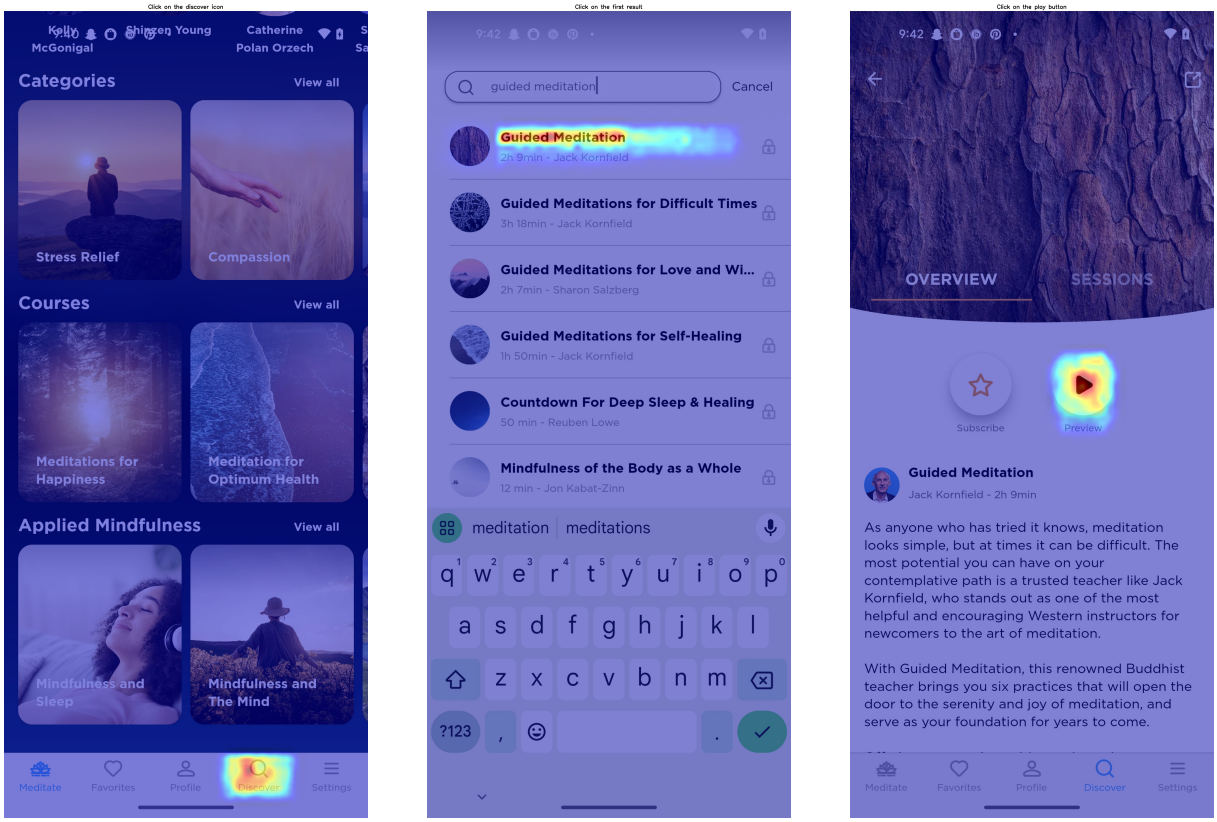


(a) Multi step grounding case 1: "Open Phase of the moon App, select the date 25 July on the calendar and view the moon phase for that date." Step 1 (left) and Step 2 (right).



(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."