

V2P: VISUAL ATTENTION CALIBRATION FOR GUI GROUNDING VIA BACKGROUND SUPPRESSION AND CENTER PEAKING

Anonymous authors
Paper under double-blind review

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 92.3% and 50.5% 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 (see Sec. 4.3), 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

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068

Instruction: "Close the Apple.com homepage tab in the Safari browser."

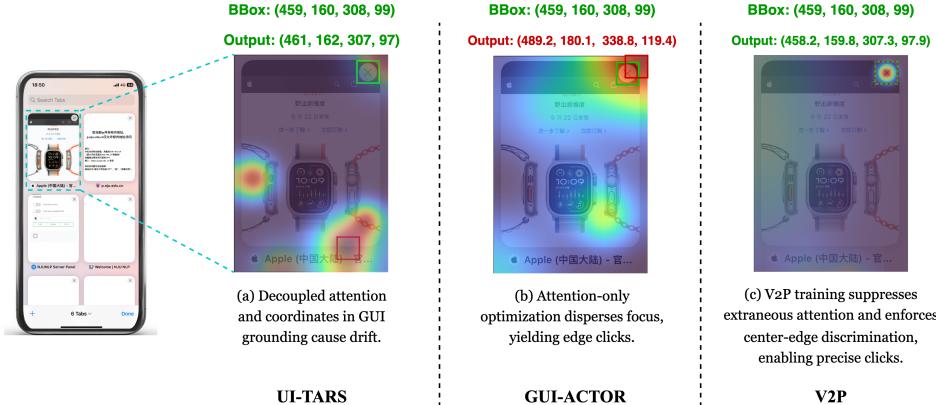


Figure 1: Comparison of different strategies in the GUI grounding task. The green box marks the ground-truth bounding box, and the red box highlights the region where the model places the highest attention given the instruction and screenshot. The overlaid heatmap is colour-coded from cool (blue) to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

075 allows these regions to absorb probability mass, weakening or even shifting the intended
076 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 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 92.3% on ScreenSpot-v2 and 50.5% on the challenging ScreenSpot-Pro.

108 with relative improvements of 3.5% and 23.7%. Furthermore, we confirm that V2P demonstrates
 109 significant practical value for real-world deployment and seamless integration into GUI agents.
 110

111

112 2 RELATED WORK

113

114

2.1 GUI-AGENTS

115

116 GUI agents have progressed from rudimentary random- or rule-based test tools to multimodal,
 117 LLM-driven systems that can follow natural-language instructions. Early efforts such as Monkey
 118 testing (Wetzlmaier et al., 2016) and planning or script record-and-replay frameworks (Memon
 119 et al., 2001; Steven et al., 2000) provided basic coverage but required hand-crafted rules or scripts.
 120 Machine-learning techniques later enabled more adaptive behaviour: Humanoid (Li et al., 2020)
 121 and Deep GUI (YazdaniBanafsheDaragh & Malek, 2022) learned user-like action policies from
 122 screenshots, while widget detectors (White et al., 2019) improved element recognition. Natural-
 123 language interfaces soon followed, e.g. FLIN (Mazumder & Riva, 2021) and RUSS (Xu et al.,
 124 2021), and reinforcement learning environments like WoB (Shi et al., 2017) and WebShop (Yao et al.,
 125 2023) pushed web-scale interaction. The recent arrival of LLMs has unified perception, reasoning
 126 and control: WebAgent (Gur et al., 2024) and WebGUM (Furuta et al., 2024) achieve open-world
 127 browsing, AutoDroid (Wen et al., 2024) and AppAgent (Zhang et al., 2023) automate smartphones,
 128 and desktop agents such as UFO (Zhang et al., 2024) demonstrate GPT-4-level capabilities; industrial
 129 systems (e.g. Claude 3.5 Sonnet and Operator) further attest to the practical traction of GUI agents.
 130

131

2.2 GUI GROUDING

132

133 Early works on GUI grounding treated it as a coordinate regression task (Zhang et al., 2025; Qin
 134 et al., 2025). However, modern methods have largely shifted to leveraging the cross-modal attention
 135 maps of Vision-Language Models (VLMs) (Cheng et al., 2024; Wu et al., 2025). In this paradigm,
 136 the model’s prediction is derived from the image patch with the highest attention score in response to
 137 a language command. While more robust, this approach often suffers from imprecise attention, with
 138 focus leaking into irrelevant background regions or spreading too uniformly across the target element.
 139 Our work directly addresses this by refining the quality of the attention map itself.

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

147

148 3 METHOD

149

150 We introduce Valley-to-Peak (V2P), a method that reshapes the model’s attention landscape to mimic
 151 human focus patterns for precise GUI grounding. It achieves this through two synergistic components:
 152

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

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

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

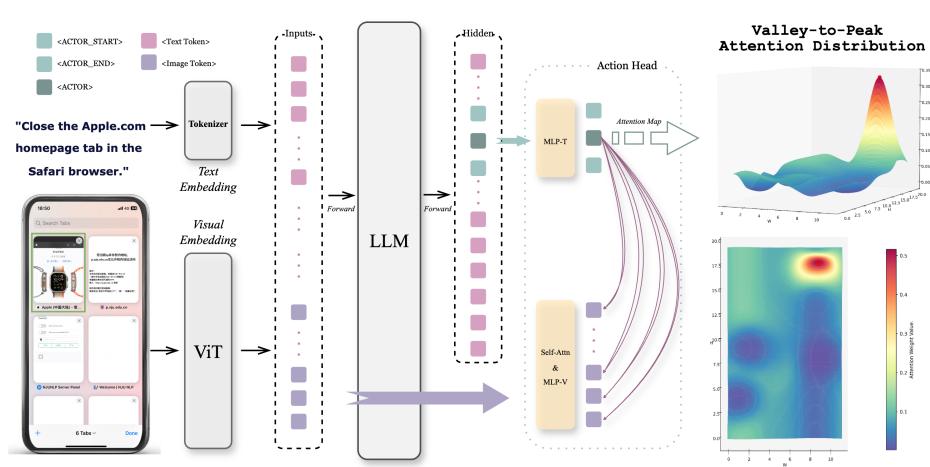


Figure 2: **Valley-to-Peak training method (V2P).** V2P jointly suppresses noise and enhances signals via two strategies: An inverse-attention penalty carves valleys in non-target areas, while size-adaptive Fitts-Gaussian peaks create sharp peaks at UI elements’ centers. This dual approach reshapes attention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

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 $\langle\text{ACTOR}\rangle$ in the output sequence as a contextual anchor. The final-layer hidden state of $\langle\text{ACTOR}\rangle$, denoted $h_{\langle\text{ACTOR}\rangle}$, is used to compute action attention over image patch features $\{v_1, \dots, v_M\}$ extracted by the vision encoder.

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

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

yielding contextualized representations. These are projected into a shared embedding space with $h_{\langle\text{ACTOR}\rangle}$ via separate MLPs:

$$z = \text{MLP}_T(h_{\langle\text{ACTOR}\rangle}), \quad (2)$$

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

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

$$\mathcal{G} = \{i \in \{1, \dots, M\} \mid R_i \cap b = \emptyset\}. \quad (5)$$

216 We define the attention loss as the total attention mass over these irrelevant regions:
 217

$$218 \quad \mathcal{L}_{\text{Attn}} = \sum_{i \in \mathcal{G}} a_i. \quad (6)$$

220 To better understand the theoretical foundation of this constraint, we analyze the gradient dynamics
 221 of attention weights. For the target patch k with attention weight $A_k = \text{softmax}(s_k)$, the gradient
 222 with respect to any non-target patch logit s_i is:
 223

$$224 \quad w_i = \frac{\partial A_k}{\partial s_i} = \frac{\partial \text{softmax}(s_k)}{\partial s_i} = -\frac{e^{s_k} e^{s_i}}{(\sum_i^M e^{s_i})^2} = -A_k A_i < 0 \quad (i \neq k). \quad (7)$$

226 This gradient analysis reveals that any increase in attention logits s_i for non-target patches negatively
 227 impacts the target attention A_k . The magnitude $|w_i| = A_k A_i$ quantifies this negative influence: larger
 228 values indicate that even small increases in attention to patch i will cause rapid degradation in target
 229 attention A_k . This theoretical insight naturally motivates using $|w_i|$ as a weighting factor in our
 230 suppression loss, providing stronger penalties for patches that pose greater threats to target attention
 231 focus. And we have the *suppression attention loss* combined with gradient weight as:
 232

$$233 \quad \mathcal{L}_{\text{Sup_Attn}} = \sum_{i \in \mathcal{G}} w_i a_i. \quad (8)$$

235 This loss encourages the model to suppress attention on irrelevant regions, thereby reducing the
 236 impact of distracting elements in cluttered interfaces. By explicitly minimizing $\mathcal{L}_{\text{Sup_Attn}}$, the model
 237 is incentivized to concentrate its focus on the target region, resulting in enhanced spatial precision
 238 and improved robustness.
 239

240 3.3 FITTS-GAUSSIAN PEAK MODELING FOR CENTER-FOCUSED GROUNDING

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

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

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

$$252 \quad \mu = (c_x, c_y) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right). \quad (9)$$

254 To reflect the interaction tolerance associated with target size, we set the standard deviation of the
 255 Gaussian proportional to the element's width and height:
 256

$$257 \quad \sigma_x = \frac{w}{\sigma_{\text{factor}}}, \quad \sigma_y = \frac{h}{\sigma_{\text{factor}}}, \quad (10)$$

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

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

$$265 \quad y_i = \int_{R_i} \mathcal{N}(x, y; \mu, \Sigma) dx dy, \quad (11)$$

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

$$270 \quad y_i = [\Phi(x_{\max}^i; c_x, \sigma_x) - \Phi(x_{\min}^i; c_x, \sigma_x)] \cdot [\Phi(y_{\max}^i; c_y, \sigma_y) - \Phi(y_{\min}^i; c_y, \sigma_y)], \quad (12)$$

270 with $\Phi(\cdot; \mu, \sigma)$ denoting the CDF of a univariate normal distribution.
 271

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

$$275 \quad \mathcal{L}_{\text{Action_Attn}} = \sum_{i=1}^M p_i \log \frac{p_i}{a_i}, \quad p_i = \frac{y_i}{\sum_{j=1}^M y_j + \epsilon}, \quad i = 1, \dots, M, \quad (13)$$

279 where ϵ is a small constant for numerical stability.
 280

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

285 3.4 VALLEY-TO-PEAK TRAINING

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

$$289 \quad \mathcal{L} = \mathcal{L}_{\text{NTP}} + \lambda_1 \mathcal{L}_{\text{Sup_Attn}} + \lambda_2 \mathcal{L}_{\text{Action_Attn}}, \quad (14)$$

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

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

300 4 EXPERIMENT

302 4.1 EXPERIMENT SETUP

304 **Setup.** We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025) as our backbone and train it on 0.7M
 305 filtered GUI screenshots, with a learning rate of 5e-6 and Gaussian factor $\sigma=1$. We evaluate on
 306 ScreenSpot-v2 (Wu et al., 2024b) and the more challenging ScreenSpot-Pro (Li et al., 2025) bench-
 307 marks using Element Accuracy. Comprehensive implementation details, including the data filtering
 308 process, are provided in App. A and B.

309 4.2 MAIN RESULT

311 Our proposed **V2P-7B** demonstrates outstanding performance across diverse benchmarks, showcasing
 312 robust generalization and superior efficiency. On the highly challenging ScreenSpot-Pro benchmark,
 313 which serves as a strong indicator of out-of-distribution (OOD) generalization, V2P-7B achieves an
 314 average accuracy of 50.54% (Tab. 1). This result significantly outperforms all GUI-specific models,
 315 including strong RL-based methods like SE-GUI-7B (47.3%) and GUI-G²-7B (47.5%). Remarkably,
 316 our 7B model even surpasses the much larger 72B-parameter UI-TARS-72B (38.1%), highlighting
 317 exceptional parameter efficiency. This strong performance is consistent across diverse scenarios,
 318 with our model securing top scores in 6 of 12 task categories and demonstrating stable adaptability
 319 in specialized domains like CAD, Creative, and Science. Furthermore, V2P-7B also excels on
 320 the ScreenSpot-v2 benchmark with an average accuracy of 92.3% and we report the result in the
 321 Appendix (See Tab. 5).

322 These advancements are driven by our dual-optimization strategy: *Suppression Attention* mitigates
 323 background distractions, while *Fitts-Gaussian Labeling* resolves center-edge confusion. This strong
 324 performance is achieved via supervised fine-tuning (SFT) alone, which highlights the potential for

further enhancements through reinforcement learning (RL) integration. The stability of our SFT approach is further evidenced by the model’s training trajectory on ScreenSpot-Pro (Fig. 4(c)), which shows no signs of persistent overfitting, unlike baselines that exhibit a continued performance decline. The consistent gains across diverse UI platforms and interaction types affirm V2P’s robust generalizability for real-world GUI grounding applications.

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.	Avg.
<i>Proprietary Models</i>																
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	
<i>General Open-source Models</i>																
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	
<i>GUI-specific Models (SFT)</i>																
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
<i>GUI-specific Models (RL)</i>																
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	
<i>Ours</i>																
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	50.54	

Table 1: 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). Further details are provided in App. C.

4.3 ATTENTION MAP QUALITY ANALYSIS

To diagnose common failure modes in GUI grounding, we manually analyzed the attention quality of 100 randomly sampled cases across our V2P model and two representative baselines (UI-TARS (Qin et al., 2025) and GUI-Actor (Wu et al., 2025)). Our analysis focused on two critical issues: **background distraction** (attention on irrelevant regions) and **center-edge confusion** (imprecise localization at the element’s boundary).

The results, summarized in Table 2, reveal a clear discrepancy in attention quality. Traditional textual-output models like UI-TARS suffer from near-total background distraction (100 cases), indicating that coordinate supervision fails to teach visual focus. While vision-attention models like GUI-Actor show improvement (74 total issues), they still struggle with background distraction and center-edge confusion. In contrast, our V2P model demonstrates superior performance, reducing background distraction to only 42 cases and center-edge confusion to 15. With a total of just 57 issues, V2P significantly outperforms both baselines, providing direct evidence that its explicit design effectively remedies these common failure modes for more reliable GUI grounding.

Attention Issue	UI-TARS	GUI-Actor	V2P (Ours)
Background distraction	100	53	42
Centre-edge confusion	0	21	15
Total Issues	100	74	57

Table 2: Attention Map Quality Analysis on 100 Manually Sampled Cases.

4.4 ABLATION STUDY

4.4.1 ABLATION STUDY FOR V2P

Our ablation study (Tab. 3a) on the challenging ScreenSpot-Pro benchmark validates the efficacy of our V2P method. Removing *Fitts-Gaussian Peak Modeling* and *Suppression Attention* individually causes performance drops of 3.0% (to 47.5%) and 3.2% respectively, highlighting their roles in resolving center-edge confusion and reducing background distractions.

On the simpler ScreenSpot-v2, removing *Fitts-Gaussian Peak Modeling* alone has a negligible impact (92.3% accuracy), as its simple layouts with minimal overlap diminish the need for precise center-point guidance. We further demonstrate in Sec. 4.4.2 that this component excels on complex, overlapping interfaces. However, removing both components still results in a slight drop to 91.9%. This shows that while V2P’s full potential is most evident in challenging scenarios like ScreenSpot-Pro, it remains robust across different complexities.

ScreenSpot-Pro	
Model	Avg.
V2P-7B (Full)	50.5
w/o FGPM	47.5 (−3.0↓)
w/o FGPM & SA	44.3 (−6.2↓)
ScreenSpot-v2	
Model	Avg.
V2P-7B (Full)	92.3
w/o FGPM	92.3 (−0.0)
w/o FGPM & SA	91.9 (−0.4↓)

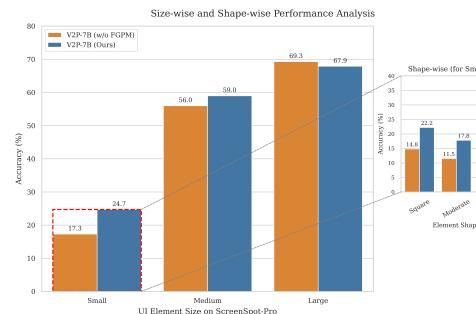
(a) Ablation study on *ScreenSpot-Pro* and *ScreenSpot-v2*.(b) Ablation study demonstrating the effectiveness of **Fitts-Gaussian Peak Modeling**.

Figure 3: Combined ablation studies. (a) Performance on different datasets. (b) Detailed breakdown for UI element size and shape.

4.4.2 ABLATION STUDY FOR EFFECTIVENESS OF FITTS-GAUSSIAN PEAK MODELING

Traditional attention methods often yield overly broad regions, misaligning with small UI elements and producing points outside their boxes (Fig. 1(b)). Fitts-Gaussian Peak Modeling counters this by centering the attention, boosting accuracy on tiny elements. We conduct ablation studies on the challenging ScreenSpot-Pro dataset (Li et al., 2025) to validate our approach.

We first split UI elements into **small**, **medium**, and **large** categories based on bounding box sizes. Fig. 3b shows that our Fitts-Gaussian Peak Modeling (FGPM) yields substantial improvements on challenging smaller elements: 7.4% for **small** and 3.0% for **medium** elements. For **large** elements, there is a slight decrease, as original attention-based methods with dispersed attention may accidentally fall within large bounding boxes even when localizing incorrectly, while our precise targeting reduces such coincidental hits.

We further analyze shape impact by categorizing **small** elements into **square**, **moderate**, and **flat** shapes based on aspect ratios. The zoomed-in table in Fig. 3b demonstrates consistent improve-

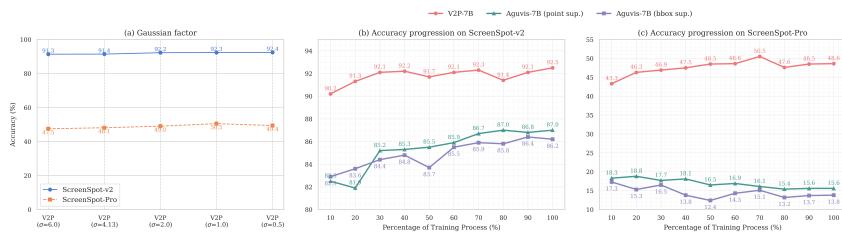


Figure 4: Ablation Study of Fitts-Gaussian Peak Modeling and Generalization Analysis of V2P. The table shows the performance impact of our proposed method and its generalization capability on an out-of-distribution dataset. Results for Aguvis-7B are from GUI-Actor (Wu et al., 2025).

ments across all shapes, confirming that FGPM effectively addresses precise localization challenges regardless of element shape.

4.4.3 ABLATION STUDY FOR GAUSSIAN FACTOR σ

We conducted ablation experiments to analyze the effect of different Gaussian factors σ on model performance. As shown in Fig. 4(a), the model's performance is strongly influenced by the choice of Gaussian factor σ . For both ScreenSpot-v2 and ScreenSpot-Pro, accuracy improves as σ decreases. For example, on ScreenSpot-v2, the accuracy rises from 91.3% at $\sigma = 6.0$ to 92.3% at $\sigma = 0.5$, while ScreenSpot-Pro achieves its best result of 50.5% accuracy at $\sigma = 1.0$.

We suspect that this is because that larger σ values correspond to a broader Gaussian distribution, which tends to dilute the spatial focus and introduce noise into the attention maps. In comparison, smaller σ produces sharper Gaussian peaks, allowing the model to localize UI elements with higher precision and resulting in more accurate click predictions. These results underscore that carefully balancing the Gaussian factor is crucial: excessively large values hinder localization, while moderate to smaller values significantly enhance spatial accuracy and overall model performance.

4.5 QUALITATIVE AND ADVANCED CAPABILITIES ANALYSIS

To provide deeper insights beyond quantitative metrics, we conducted a series of qualitative and advanced capability analyses, with full details and visualizations provided in App. D.

Our qualitative review (Fig. 5 and 6) confirms that V2P generates sharp, well-defined attention maps that align closely with target element boundaries, successfully mitigating common failure modes like semantic confusion and low-confidence predictions. Furthermore, we validated V2P's practical utility in more complex scenarios. As shown in Fig. 7(a) and 8, the model demonstrates robust performance in **multi-step interaction** workflows, maintaining contextual awareness across sequential operations. It also exhibits sophisticated **multi-target localization** capabilities in Fig. 7(b), simultaneously identifying multiple elements within a single interface.

Finally, we integrated V2P into an end-to-end agent to tackle a real-world, multi-app task. The model successfully completed the entire 7-step trajectory without error (Fig. 10), confirming its potential as a reliable grounding component for practical GUI automation.

5 CONCLUSION

We presented V2P, a novel framework for GUI grounding that operationalizes a "valley-to-peak" strategy. By first suppressing background distractions and then highlighting clickable regions with a Fitts-Gaussian peak, V2P explicitly addresses the critical issues of background distraction and center-edge confusion. Our approach achieves state-of-the-art performance, attaining 92.3% accuracy on ScreenSpot-v2 and 50.5% on the challenging ScreenSpot-Pro benchmark. Extensive experiments validate the effectiveness of each component and demonstrate the strong generalization capabilities of our model. As a lightweight, interpretable, and scalable solution, V2P offers tangible benefits for developing robust GUI agents capable of operating in complex, real-world software environments.

486 REPRODUCIBILITY STATEMENT
487488 We have made every effort to ensure the reproducibility of our results. Key experimental settings,
489 datasets, model architectures, and training procedures are described in detail in the main paper
490 (Sec. 4.1 and App. A). The anonymous source code is available via a link in the supplementary
491 materials. We encourage readers to consult these resources for detailed replication guidance.
492493 ACKNOWLEDGEMENT
494495 During the preparation of this manuscript, we used Google Gemini-2.5-Pro (gem, 2025) to assist with
496 language polishing and proofreading to improve the clarity and readability of the text. The authors
497 assume full responsibility for the final content.
498499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

540 REFERENCES

541

542 Google team, 2025. URL <https://arxiv.org/abs/2312.11805>.

543 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang,
544 Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan,
545 Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng,
546 Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-v1 technical report. *arXiv*
547 preprint *arXiv:2502.13923*, 2025.

548 Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Shuai Ren, and
549 Hongsheng Li. Amex: Android multi-annotation expo dataset for mobile gui agents, 2025. URL
550 <https://arxiv.org/abs/2407.17490>.

551 Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu,
552 Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse:
553 From general vision language models to versatile gui agents, 2025. URL <https://arxiv.org/abs/2406.11317>.

554 Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu.
555 SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd*
556 *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.
557 9313–9332, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL
558 <https://aclanthology.org/2024.acl-long.505>.

559 Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude
560 of movement. *Journal of experimental psychology*, 47 6:381–91, 1954. URL <https://api.semanticscholar.org/CorpusID:501599>.

561 Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane
562 Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models,
563 2024. URL <https://arxiv.org/abs/2305.11854>.

564 Google. Claude 3.5 sonnet model card addendum. In *Claude 3.5 Sonnet Model Card Addendum*,
565 2024. URL <https://api.semanticscholar.org/CorpusID:270667923>.

566 Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and
567 Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents,
568 2025a. URL <https://arxiv.org/abs/2410.05243>.

569 Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun,
570 and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI
571 agents. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL
572 <https://openreview.net/forum?id=kxnoqaisCT>.

573 Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and
574 Aleksandra Faust. A real-world webagent with planning, long context understanding, and program
575 synthesis, 2024. URL <https://arxiv.org/abs/2307.12856>.

576 Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan
577 Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A
578 visual language model for gui agents, 2024. URL <https://arxiv.org/abs/2312.08914>.

579 Kaixin Li, Meng Ziyang, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang,
580 and Tat-Seng Chua. Screenspot-pro: GUI grounding for professional high-resolution computer
581 use. In *Workshop on Reasoning and Planning for Large Language Models*, 2025. URL <https://openreview.net/forum?id=XaKNDAHas>.

582 Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided
583 attention inference network, 2018. URL <https://arxiv.org/abs/1802.10171>.

584 Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu,
585 and Oriana Riva. On the effects of data scale on ui control agents, 2024. URL <https://arxiv.org/abs/2406.03679>.

594 Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. Humanoid: A deep learning-based
 595 approach to automated black-box android app testing, 2020. URL <https://arxiv.org/abs/1901.02633>.
 596

597 Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei,
 598 Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual
 599 agent, 2024. URL <https://arxiv.org/abs/2411.17465>.
 600

601 Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang,
 602 and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative
 603 reasoners, 2025. URL <https://arxiv.org/abs/2504.14239>.
 604

605 Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren,
 606 Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by
 607 reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.21620>.
 608

609 Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1 : A generalist r1-style vision-language
 610 action model for gui agents, 2025. URL <https://arxiv.org/abs/2504.10458>.
 611

612 I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-*
613 Comput. Interact., 7(1):91–139, March 1992. ISSN 0737-0024. doi:10.1207/s15327051hci0701_3.
 614

615 URL https://doi.org/10.1207/s15327051hci0701_3.
 616

617 Sahisnu Mazumder and Oriana Riva. Flin: A flexible natural language interface for web navigation,
 618 2021. URL <https://arxiv.org/abs/2010.12844>.
 619

620 A.M. Memon, M.E. Pollack, and M.L. Soffa. Hierarchical gui test case generation using
 621 automated planning. *IEEE Transactions on Software Engineering*, 27(2):144–155, 2001.
 622 doi:10.1109/32.908959.
 623

624 OpenAI. OpenAI Operator, 2023. URL <https://github.com/openai-operator/openai-operator>. Accessed: 2023-10-13.
 625

626 OpenAI. Gpt-4o system card, 2024. URL <https://arxiv.org/abs/2410.21276>.
 627

628 Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang,
 629 Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin,
 630 Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng,
 631 Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng
 632 Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. Ui-tars: Pioneering automated gui interaction
 633 with native agents, 2025. URL <https://arxiv.org/abs/2501.12326>.
 634

635 Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits:
 636 An open-domain platform for web-based agents. In Doina Precup and Yee Whye Teh (eds.),
 637 *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings
 638 of Machine Learning Research*, pp. 3135–3144. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/shi17a.html>.
 639

640 John Steven, Pravir Chandra, Bob Fleck, and Andy Podgurski. jrapture: A capture/replay tool for
 641 observation-based testing. *SIGSOFT Softw. Eng. Notes*, 25(5):158–167, August 2000. ISSN 0163-
 642 5948. doi:10.1145/347636.348993. URL <https://doi.org/10.1145/347636.348993>.
 643

644 Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang,
 645 Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueling Zhuang. Gui-g²: Gaussian
 646 reward modeling for gui grounding, 2025a. URL <https://arxiv.org/abs/2507.15846>.
 647

648 Jiaqi Tang, Yu Xia, Yi-Feng Wu, Yuwei Hu, Yuhui Chen, Qing-Guo Chen, Xiaogang Xu, Xiangyu Wu,
 649 Hao Lu, Yanqing Ma, Shiyin Lu, and Qifeng Chen. Lpo: Towards accurate gui agent interaction via
 650 location preference optimization, 2025b. URL <https://arxiv.org/abs/2506.09373>.
 651

652 Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai,
 653 Cong Yao, and Zhibo Yang. Omniparser: A unified framework for text spotting, key information
 654 extraction and table recognition, 2024. URL <https://arxiv.org/abs/2403.19128>.
 655

648 Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao
 649 Liu, Yaqin Zhang, and Yunxin Liu. Autodroid: Llm-powered task automation in android, 2024.
 650 URL <https://arxiv.org/abs/2308.15272>.

651 Thomas Wetzlmaier, Rudolf Ramler, and Werner Putschögl. A framework for monkey gui testing. In
 652 *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pp.
 653 416–423, 2016. doi:10.1109/ICST.2016.51.

654 Thomas D. White, Gordon Fraser, and Guy J. Brown. Improving random gui testing with image-based
 655 widget detection. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software
 656 Testing and Analysis*, ISSTA 2019, pp. 307–317, New York, NY, USA, 2019. Association for
 657 Computing Machinery. ISBN 9781450362245. doi:10.1145/3293882.3330551. URL <https://doi.org/10.1145/3293882.3330551>.

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

661 Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng,
 662 Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: A foundation action model for
 663 generalist gui agents, 2024a. URL <https://arxiv.org/abs/2410.23218>.

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

667 Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu,
 668 Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and
 669 Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis,
 670 2025. URL <https://arxiv.org/abs/2505.13227>.

671 Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James Landay, and Monica S
 672 Lam. Grounding open-domain instructions to automate web support tasks, 2021. URL <https://arxiv.org/abs/2103.16057>.

673 Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu,
 674 and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction, 2025.
 675 URL <https://arxiv.org/abs/2412.04454>.

676 Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai,
 677 Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. Magma: A foundation
 678 model for multimodal ai agents, 2025a. URL <https://arxiv.org/abs/2502.13130>.

679 Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-
 680 ui: Visual grounding for gui instructions, 2025b. URL <https://arxiv.org/abs/2412.16256>.

681 Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable
 682 real-world web interaction with grounded language agents, 2023. URL <https://arxiv.org/abs/2207.01206>.

683 Faraz YazdaniBanafsheDaragh and Sam Malek. Deep gui: black-box gui input generation with
 684 deep learning. In *Proceedings of the 36th IEEE/ACM International Conference on Automated
 685 Software Engineering*, ASE '21, pp. 905–916. IEEE Press, 2022. ISBN 9781665403375.
 686 doi:10.1109/ASE51524.2021.9678778. URL <https://doi.org/10.1109/ASE51524.2021.9678778>.

687 Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin
 688 Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. Enhancing visual grounding for gui agents via self-
 689 evolutionary reinforcement learning, 2025. URL <https://arxiv.org/abs/2505.12370>.

690 Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei
 691 Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Ufo: A ui-focused agent for windows os
 692 interaction, 2024. URL <https://arxiv.org/abs/2402.07939>.

702 Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma,
703 Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large language
704 model-brained gui agents: A survey, 2025. URL <https://arxiv.org/abs/2411.18279>.

705
706 Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu.
707 Appagent: Multimodal agents as smartphone users, 2023. URL <https://arxiv.org/abs/2312.13771>.

709 Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding
710 r1-zero-like training for visual grounding in gui agents, 2025. URL <https://arxiv.org/abs/2505.15810>.

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 **A TRAINING AND INFERENCE DETAILS**
757758 **A.1 SOURCE TRAINING DATA**
759

760 Following GUI-Actor (Wu et al., 2025), we compile our training dataset from several publicly
761 available, high-quality GUI datasets, with summary statistics provided in Table 3. To ensure fair
762 evaluation, we also exclude any samples from Wave-UI that overlap with the test sets of downstream
763 tasks. Our data recipe is built from several public GUI datasets, the source data totaling approximately
764 1M screenshots. To ensure annotation quality, we apply Ominiparser (Wan et al., 2024) to detect
765 bounding boxes for all samples and filter those where the IoU between ground truth (GT) and parser-
766 detected boxes is less than 0.3, as such cases likely contain annotation errors, this step improves the
767 data consistency for training. After filtering, there are about ~ 0.7 M screenshots remains.
768

769 Dataset	770 # of Elements	771 # of Screenshots	772 Platform
770 Uground Web-Hybrid (Gou et al., 2025a)	771 8M	775K	776 Web
771 GUI-Env (Chen et al., 2025)	772 262K	776 70K	777 Web
772 GUI-Act (Chen et al., 2025)	773 42K	777 13K	778 Web
773 AndroidControl (Li et al., 2024)	774 47K	778 47K	779 Android
774 AMEX (Chai et al., 2025)	775 1.2M	779 100K	780 Android
775 Wave-UI	776 50K	780 7K	781 Hybrid
Total		9.6M	–

778 Table 3: Overview of training datasets used for GUI-Actor.
779780 **A.2 TRAINING AND INFERENCE SETUP**
781

782 During the training phase, we first freeze the backbone VLM parameters and train only the action
783 head (~ 20 M parameters). In the second phase, we fine-tune the entire model using the filtered
784 dataset with standard supervised learning. At inference, we follow deterministic generation with a
785 temperature of 0 and adopt a confidence threshold of $\gamma = 0.95$ for the ScreenSpot-Pro benchmark
786 and $\gamma = 0.8$ for ScreenSpot-v2 tasks.
787

788 **A.3 TRAINING AND INFERENCE COSTS**
789790 **A.3.1 TRAINING COSTS**
791

792 The V2P-7B model was trained using a comprehensive dataset comprising Uground Web-Hybrid,
793 GUI-Env, GUI-Act, AndroidControl, AMEX, and Wave-UI (Tab. 3). The training process was
794 conducted on a high-performance computing cluster equipped with 32 NVIDIA H200 GPUs.

795 Following the (Wu et al., 2025), the training procedure consisted of two phases: an initial warmup
796 phase which only fine-tunes the parameters of attention heads requiring approximately 15 hours,
797 followed by full supervised fine-tuning (SFT) that took an additional 20 hours. This results in a total
798 training time of approximately 35 hours across 32 H200 GPUs, equivalent to 1,120 GPU-hours for
799 the complete training pipeline.
800

801 **A.3.2 INFERENCE EFFICIENCY**
802

803 Tab. 4 presents the detailed inference performance metrics of the V2P-7B model evaluated on different
804 benchmarks with batch size 1.
805

805 Metric	806 ScreenSpot-v2	807 ScreenSpot-Pro
806 Latency per Sample (ms)	807 154.00	808 1857.97
807 Throughput (samples/sec)	808 6.49	809 0.54

809 Table 4: Inference Performance Metrics

810
811 The model demonstrates efficient inference performance, with significantly faster processing on
812 ScreenSpot-v2 compared to ScreenSpot-Pro, likely due to the complexity differences between the
813 two benchmarks.

814 **A.3.3 RESOURCE REQUIREMENTS**

815
816 The V2P-7B model comprises approximately 7 billion parameters and requires around 110 GB of
817 disk storage for the model weights. During inference, the model consumes approximately 80 GB of
818 GPU memory at peak usage. All inference evaluations were conducted on a single NVIDIA H200
819 GPU with 141 GB of memory, demonstrating that the model can be efficiently deployed on high-end
820 consumer or enterprise-grade hardware.

821 The computational requirements make the model accessible for research and production environments
822 with sufficient GPU resources, while the inference speeds are suitable for real-time applications,
823 particularly on the ScreenSpot-v2 benchmark.

824 **B BENCHMARKS**

825
826 Our evaluation centers on two sophisticated benchmarks for GUI visual grounding: ScreenSpot-
827 v2 (Wu et al., 2024b) and ScreenSpot-Pro (Li et al., 2025).

828
829 **ScreenSpot-v2** encompasses 1,272 carefully annotated instructions, each paired with corresponding
830 target elements across diverse GUI environments, including mobile (Android and iOS), desktop
831 (macOS and Windows), and web platforms. The dataset is designed to improve the quality and
832 reliability of GUI visual grounding tasks, addressing key challenges such as eliminating ambiguities
833 in natural language instructions and resolving annotation errors. By refining the alignment between
834 textual descriptions and interface elements, ScreenSpot-v2 provides a robust and standardized
835 benchmark for evaluating grounding models.

836
837 **ScreenSpot-Pro**, meanwhile, focuses on more demanding scenarios, especially those involving
838 high-resolution professional applications. It contains 1,581 tasks annotated by domain experts across
839 23 specialized software applications, spanning three operating systems. This benchmark significantly
840 broadens the scope of GUI visual grounding by introducing interfaces with industrial software and
841 multi-window layouts, creating a larger domain gap compared to most pretraining data. With its
842 increased complexity and domain diversity, ScreenSpot-Pro is an invaluable resource for assessing
843 the generalization ability of models in realistic and challenging GUI environments.

844 **C BASELINES**

845
846 **C.1 BASELINES FOR SCREENSPOT-PRO**

847
848 We establish comprehensive benchmarking across four categories of state-of-the-art GUI understand-
849 ing systems:

- 850 • **Proprietary Systems:** GPT-4o (OpenAI, 2024) (vision-language foundation model), Claude
851 Computer Use (Google, 2024) (specialized GUI agent)
- 852 • **General-Purpose Open-Source:** Qwen2.5-VL series (Bai et al., 2025) (7B/72B parameter
853 variants)
- 854 • **GUI-Specialized (SFT):**
 - 855 – Medium-scale: SeeClick-9.6B (Based on Qwen-VL-Chat) (Cheng et al., 2024),
FOCUS-2B (Based on Qwen2-VL-2B-Instruct) (Zhang et al., 2024), OS-Atlas-7B
(Based on Qwen2-VL-7B) (Wu et al., 2024a)
 - 856 – Large-scale: CogAgent-18B (Based on CogVLM17B) (Hong et al., 2024), Aria-UI
(Based on Megatron-LM) (Yang et al., 2025b), JEDI series (Based on Qwen2.5-VL
series) (Xie et al., 2025)
 - 857 – Domain-specific: ShowUI-2B (Based on Qwen2-VL-2B) (Lin et al., 2024), Uground
series (Based on Qwen2-VL series) (Gou et al., 2025b), UI-TARS series (Based on
Qwen2-VL series) (Qin et al., 2025)

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	92.3

Table 5: 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.

- **GUI-Specialized (RL):**

- R1-style: UI-R1 (Based on Qwen2.5-VL-3B-Instruct) (Lu et al., 2025), GUI-R1 (Based on Qwen2.5-VL-3B and Qwen2.5-VL-7B) (Luo et al., 2025), InfiGUI-R1-3B (Based on Qwen2.5-VL-3B-Instruct) (Liu et al., 2025)
- Gaussian-based: GUI-G1-3B (Based on Qwen2.5-VL-3B-Instruct) (Zhou et al., 2025), SE-GUI (Based on Qwen2.5-VL-3B and Qwen2.5-VL-7B) (Yuan et al., 2025), GUI-G²-7B (Based on Qwen2.5-VL-7B-Instruct) (Tang et al., 2025a)

C.2 BASELINES FOR SCREENSPOT-V2

We establish comprehensive benchmarking across four categories of state-of-the-art GUI understanding systems:

- **Proprietary Systems:** Operator (OpenAI, 2023) (proprietary multimodal system), GPT-4o + OmniParser-v2 (OpenAI, 2024; Wan et al., 2024) (enhanced vision-language integration)
- **General-Purpose Open-Source:** Qwen2.5-VL series (Bai et al., 2025) (7B/72B parameter variants)
- **GUI-Specialized (SFT):**
 - Medium-scale: SeeClick-9.6B (Cheng et al., 2024), Magma-8B (Yang et al., 2025a), OS-Atlas series (Wu et al., 2024a)
 - Large-scale: UI-TARS series (Qin et al., 2025), Uground series (Gou et al., 2025b), GUI-Actor series (Wu et al., 2025)
 - Domain-specific: Aguvis-7B (Xu et al., 2025)

918 • **GUI-Specialized (RL):**
 919 – SE-GUI-based: SE-GUI-7B (Yuan et al., 2025), LPO-8B (Tang et al., 2025b)

920

921

D ADDITIONAL EXPERIMENTAL RESULTS AND ANALYSIS

923

924 D.1 QUALITATIVE ANALYSIS OF MODEL PERFORMANCE

925

926 D.1.1 SUCCESS CASES

927

928 Fig. 5 demonstrate several representative success cases where our V2P-7B model achieves accurate
 929 GUI element localization. Through these successful examples, we observe that the model exhibits
 930 high confidence in precisely highlighting target regions, with attention distributions that closely align
 931 with the actual shapes of UI elements. The attention maps show sharp, well-defined boundaries
 932 that accurately correspond to button edges, text field borders, and icon contours. This demonstrates
 933 the model’s robust understanding of visual-semantic correspondence between natural language
 934 instructions and GUI components, effectively bridging the gap between textual descriptions and
 935 visual interface elements.

936 D.1.2 FAILURE CASES AND ERROR ANALYSIS

937

938 Our analysis of failure cases reveals several interesting patterns and limitations, as illustrated in Fig. 6.
 939 In some instances, we observe that the model encounters difficulties when multiple UI elements
 940 share semantic similarities. The model often exhibits high confidence while incorrectly selecting
 941 semantically related but functionally different elements or misidentifying similar icons with different
 942 purposes (Fig. 6a).

943 Additionally, we identify cases where the model’s attention distribution becomes highly dispersed
 944 across the interface, which we interpret as an indicator of *low confidence* (Fig. 6b). This scattered
 945 attention pattern typically occurs in scenarios with numerous distracting elements or cluttered
 946 interfaces, suggesting that the model’s decision-making process becomes uncertain when faced with
 947 complex visual layouts.

948 Furthermore, we observe failure modes where the model’s attention concentrates entirely on regions
 949 completely unrelated to the target element (Fig. 6c). These cases often involve ambiguous natural
 950 language descriptions or interfaces with unconventional design patterns that deviate from the model’s
 951 training distribution. Such failures highlight the need for enhanced user intent understanding and
 952 more comprehensive UI context comprehension capabilities.

953 D.1.3 MULTI-STEP INTERACTION SCENARIOS

954

955 To evaluate the model’s capability in complex interaction workflows, we designed multi-step interac-
 956 tion scenarios using pure grounding tasks selected from the AndroidControl (Li et al., 2024) dataset.
 957 Fig. 7a and Fig. 8 showcases the model’s performance across sequential GUI operations.

958 The results demonstrate that our model maintains consistent accuracy throughout extended interaction
 959 sequences, successfully completing multi-step tasks that require contextual understanding and state
 960 awareness. This capability highlights the model’s potential for integration into automated GUI Agent
 961 frameworks, where reliable multi-step interaction is crucial for practical deployment. We conducted
 962 an experiment that incorporate our V2P-7B into an end-to-end real-world application case, more
 963 details can be seen in Appendix D.2

964

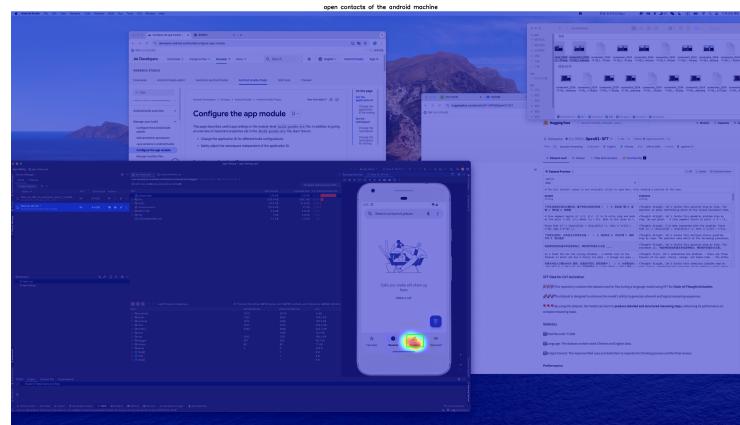
965 D.1.4 MULTI-TARGET LOCALIZATION CAPABILITIES

966

967 We investigated the model’s ability to simultaneously localize multiple targets within a single interface,
 968 which holds significant value for batch operations and improving inference efficiency. Fig. 7b presents
 969 our experimental setup using a calculator interface, where we tasked the model with simultaneously
 970 localizing the elements "1", "0", and "00".

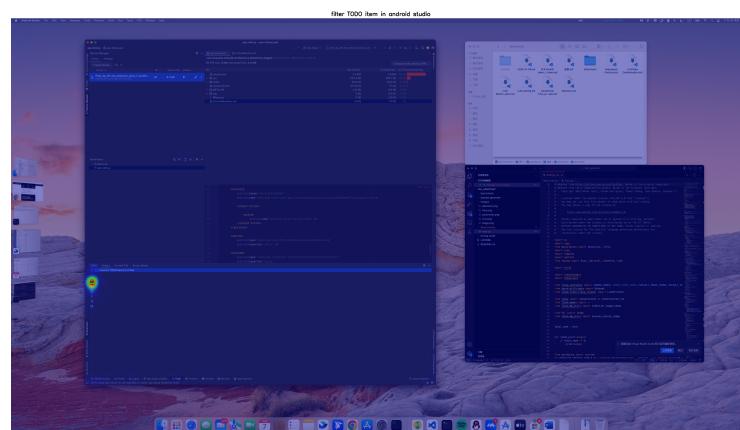
971 The results reveal that the model successfully generates attention distributions for all three target
 972 elements simultaneously, with appropriately differentiated confidence levels. Notably, the element "1"

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989



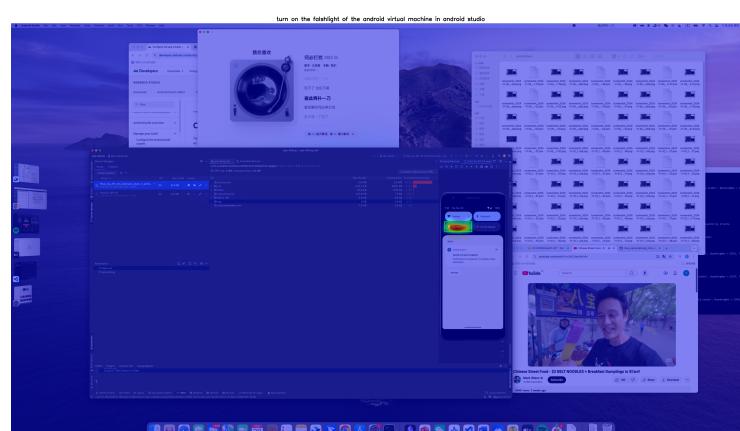
(a) Success Case 1

990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004



(b) Success Case 2

1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019

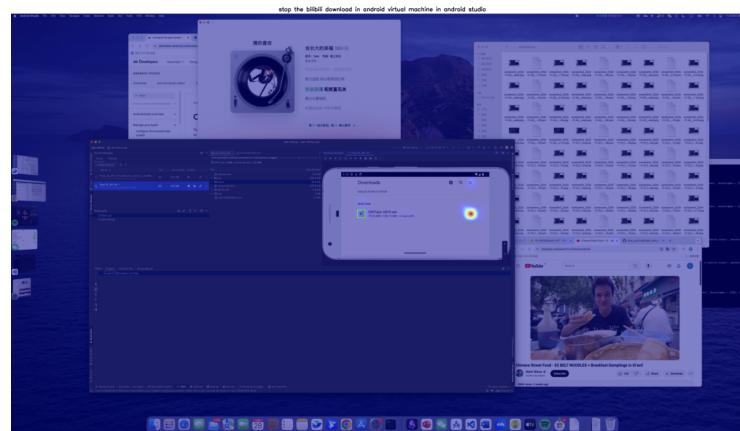


(c) Success Case 3

1020
1021
1022
1023
1024
1025

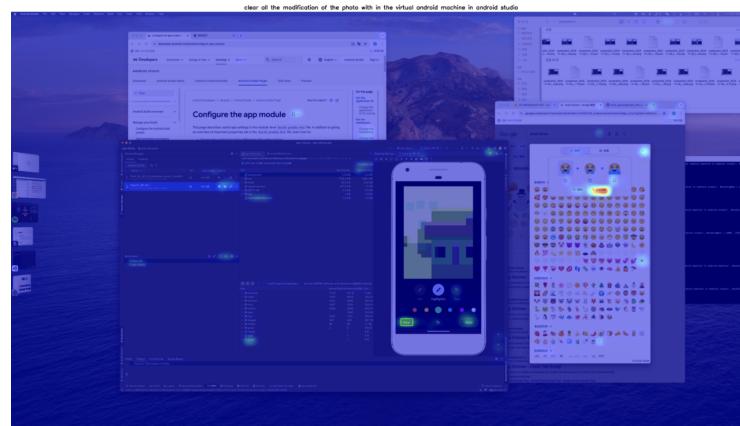
Figure 5: Representative success cases of GUI element localization.

1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043



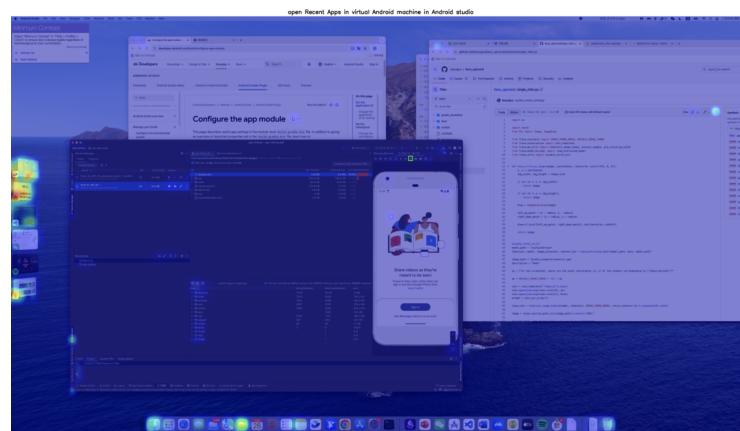
(a) Failure Case 1

1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058



(b) Failure Case 2

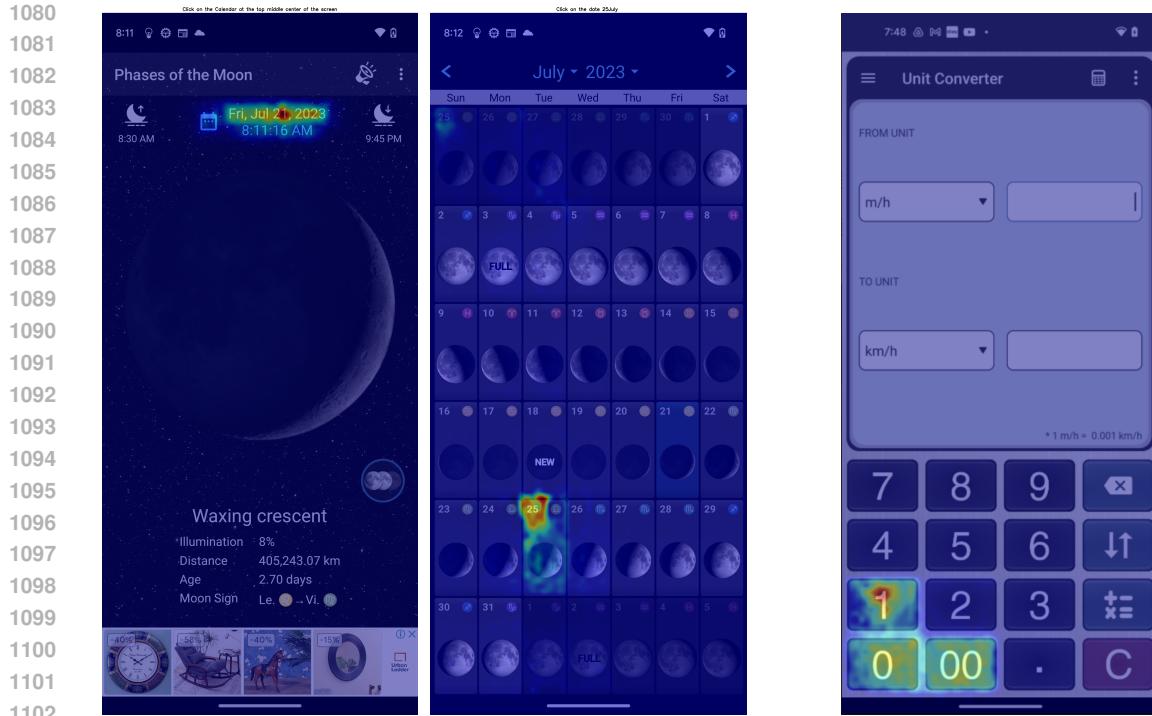
1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073



(c) Failure Case 3

1074
 1075
 1076
 1077
 1078
 1079

Figure 6: 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 7: Multi-step grounding case and multi-target grounding case.

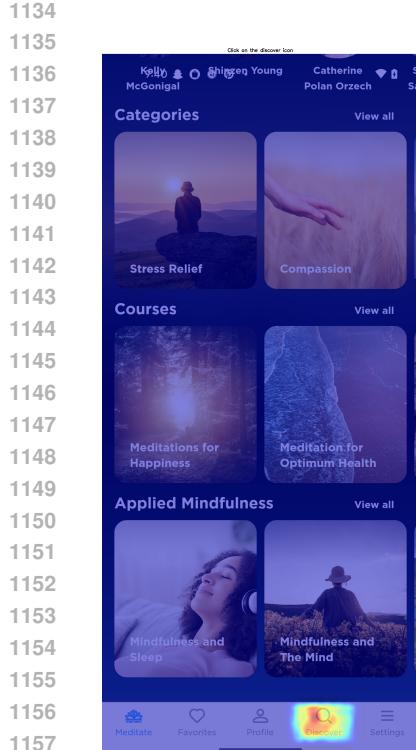
receives the highest attention intensity, followed by "0" and "00" respectively, which aligns with the natural priority and visual prominence of these elements. This multi-target capability demonstrates the model's sophisticated attention mechanism and its potential for complex GUI analysis tasks requiring simultaneous element identification, as well as its genuine understanding capability of user queries.

D.2 END-TO-END REAL-WORLD APPLICATION

D.2.1 TASK DESIGN AND SETUP

To validate the practical value of our grounding model in real-world scenarios, we selected a complex, complete GUI navigation task from the AndroidControl dataset (Li et al., 2024). The chosen task encompasses a comprehensive action space including "*navigate_back*", "*click*", and "*type_text*" operations, spanning a total of 7 sequential steps that effectively simulate realistic user interaction circumstance.

The task involves navigating through a complex multi-application workflow on mobile devices, requiring seamless transitions between different apps (Daff Moon app and Gmail app), information extraction and processing, and email composition with specific recipient details. This scenario was specifically chosen to test the model's ability to handle real-world user requests that span multiple applications, maintain context across app switches, and accurately interpret nuanced user intents involving personal relationships and specific communication requirements. We integrated our V2P-7B model into the grounding components of the navigation pipeline, replacing the baseline grounding module while maintaining the overall task execution framework.



1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
17310
17311
17312
17313
17314
17315
17316
17317
17318
17319
17320
17321
17322
17323
17324
17325
17326
17327
17328
17329
17330
17331
17332
17333
17334
17335
17336
17337
17338
17339
17340
17341
17342
17343
17344
17345
17346
17347
17348
17349
17350
17351
17352
17353
17354
17355
17356
17357
17358
17359
17360
17361
17362
17363
17364
17365
17366
17367
17368
17369
17370
17371
17372
17373
17374
17375
17376
17377
17378
17379
17380
17381
17382
17383
17384
17385
17386
17387
17388
17389
17390
17391
17392
17393
17394
17395
17396
17397
17398
17399
17400
17401
17402
17403
17404
17405
17406
17407
17408
17409
17410
17411
17412
17413
17414
17415
17416
17417
17418
17419
17420
17421
17422
17423
17424
17425
17426
17427
17428
17429
17430
17431
17432
17433
17434
17435
17436
17437
17438
17439
17440
17441
17442
17443
17444
17445
17446
17447
17448
17449
17450
17451
17452
17453
17454
17455
17456
17457
17458
17459
17460
17461
17462
17463
17464
17465
17466
17467
17468
17469
17470
17471
17472
17473
17474
17475
17476
17477
17478
17479
17480
17481
17482
17483
17484
17485
17486
17487
17488
17489
17490
17491
17492
17493
17494
17495
17496
17497
17498
17499
17500
17501
17502
17503
17504
17505
17506
17507
17508
17509
17510
17511
17512
17513
17514
17515
17516
17517
17518
17519
17520
17521
17522
17523
17524
17525
17526
17527
17528
17529
17530
17531
17532
17533
17534
17535
17536
17537
17538
17539
17540
17541
17542
17543
17544
17545
17546
17547
17548
17549
17550
17551
17552
17553
17554
17555
17556
17557
17558
17559
17560
17561
17562
17563
17564
17565
17566
17567
17568
17569
17570
17571
17572
17573
17574
17575
17576
17577
17578
17579
17580
17581
17582
17583
17584
17585
17586
17587
17588
17589
17590
17591
17592
17593
17594
17595
17596
17597
17598
17599
17600
17601
17602
17603
17604
17605
17606
17607
17608
17609
17610
17611
17612
17613
17614
17615
17616
17617
17618
17619
17620
17621
17622
17623
17624
17625
17626
17627
17628
17629
17630
17631
17632
17633
17634
17635
17636
17637
17638
17639
17640
17641
17642
17643
17644
17645
17646
17647
17648
17649
17650
17651
17652
17653
17654
17655
17656
17657
17658
17659
17660
17661
17662
17663
17664
17665
17666
17667
17668
17669
17670
17671
17672
17673
17674
17675
17676
17677
17678
17679
17680
17681
17682
17683
17684
17685
17686
17687
17688
17689
17690
17691
17692
17693
17694
17695
17696
17697
17698
17699
17700
17701
17702
17703
17704
17705
17706
17707
17708
17709
17710
17711
17712
17713
17714
17715
17716
17717
17718
17719
17720
17721
17722
17723
17724
17725
17726
17727
17728
17729
17730
17731
17732
17733
17734
17735
17736
17737
17738
17739
17740
17741
17742
17743
17744
17745
17746
17747
17748
17749
17750
17751
17752
17753
17754
17755
17756
17757
17758
17759
17760
17761
17762
17763
17764
17765
17766
17767
17768
17769
17770
17771
17772
17773
17774
17775
17776
17777
17778
17779
17780
17781
17782
17783
17784
17785
17786
17787
17788
17789
17790
17791
17792
17793
17794
17795
17796
17797
17798
17799
17800
17801
17802
17803
17804
17805
17806
17807
17808
17809
17810
17811
17812
17813
17814
17815
17816
17817
17818
17819
17820
17821
17822
17823
17824
17825
17826
17827
17828
17829
17830
17831
17832
17833
17834
17835
17836
17837
17838
17839
17840
17841
17842
17843
17844
17845
17846
17847
17848
17849
17850
17851
17852
17853
17854
17855
17856
17857
17858
17859
17860
17861
17862
17863
17864
17865
17866
17867
17868
17869
17870
17871
17872
17873
17874
17875
17876
17877
17878
17879
17880
17881
17882
17883
17884
17885
17886
17887
17888
17889
17890
17891
17892
17893
17894
17895
17896
17897
17898
17899
17900
17901
17902
17903
17904
17905
17906
17907
17908
17909
17910
17911
17912
17913
17914
17915
17916
17917
17918
17919
17920
17921
17922
17923
17924
17925
17926
17927
17928
17929
17930
17931
17932
17933
17934
17935
17936
17937
17938
17939
17940
17941
17942
17943
17944
17945
17946
17947
17948
17949
17950
17951
17952
17953
17954
17955
17956
17957
17958
17959
17960
17961
17962
17963
17964
17965
17966
17967
17968
17969
17970
17971
17972
17973
17974
17975
17976
17977
17978
17979
17980
17981
17982
17983
17984
17985
17986
17987
17988
17989
17990
17991
17992
17993
17994
17995
17996
17997
17998
17999
18000
18001
18002
18003
18004
18005
18006
18007
18008
18009
18010
18011
18012
18013
18014
18015
18016
18017
18018
18019
18020
18021
18022
18023
18024
18025
18026
18027
18028
18029
18030
18031
18032
18033
18034
18035
18036
18037
18038
18039
18040
18041
18042
18043
18044
18045
18046
18047
18048
18049
18050
18051
18052
18053
18054
18055
18056
18057
18058
18059
18060
18061
18062
18063
18064
18065
18066
18067
18068
18069
18070
18071
18072
18073
18074
18075
18076
18077
18078
18079
18080
18081
18082
18083
18084
18085
18086
18087
18088
18089
18090
18091
18092
18093
18094
18095
18096
18097
18098
18099
18100
18101
18102
18103
18104
18105
18106
18107
18108
18109
18110
18111
18112
18113
18114
18115
18116
18117
18118
18119
18120
18121
18122
18123
18124
18125
18126
18127
18128
18129
18130
18131
18132
18133
18134
18135
18136
18137
18138
18139
18140
18141
18142
18143
18144
18145
18146
18147
18148
18149
18150
18151
18152
18153
18154
18155
18156
18157
18158
18159
18160
18161
18162
18163
18164
18165
18166
18167
18168
18169
18170
18171
18172
18173
18174
18175
18176
18177
18178
18179
18180
18181
18182
18183
18184
18185
18186
18187
18188
18189
18190
18191
18192
18193
18194
18195
18196
18197
18198
18199
18200
18201
18202
18203
18204
18205
18206
18207
18208
18209
18210
18211
18212
18213
18214
18215
18216
18217
18218
18219
18220
18221
18222
18223
18224
18225
18226
18227
18228
18229
18230
18231
18232
18233
18234
18235
18236
18237
18238
18239
18240
18241
18242
18243
18244
18245
18246
18247
18248
18249
18250
18251
18252
18253
18254
18255
18256
18257
18258
18259
18260
18261
18262
18263
18264
18265
18266
18267
18268
18269
18270
18271
18272
18273
18274
18275
18276
18277
18278
18279
18280
18281
18282
18283
18284
18285
18286
18287
18288
18289
18290
18291
18292
18293
18294
18295
18296
18297
18298
18299
18300
18301
18302
18303
18304
18305
18306
18307
18308
18309
18310
18311
18312
18313
18314
18315
18316
18317
18318
18319
18320
18321
18322
18323
18324
18325
18326
18327
18328
18329
18330
18331
18332
18333
18334
18335
18336
18337
18338
18339
18340
18341
18342
18343
18344
18345
18346
18347
18348
18349
18350
18351
18352
18353
18354
18355
18356
18357
18358
18359
18360
18361
18362
18363
18364
18365
18366
18367
18368
18369
18370
18371
18372
18373
18374
18375
18376
18377
18378
18379
18380
18381
18382
18383

```

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231

```

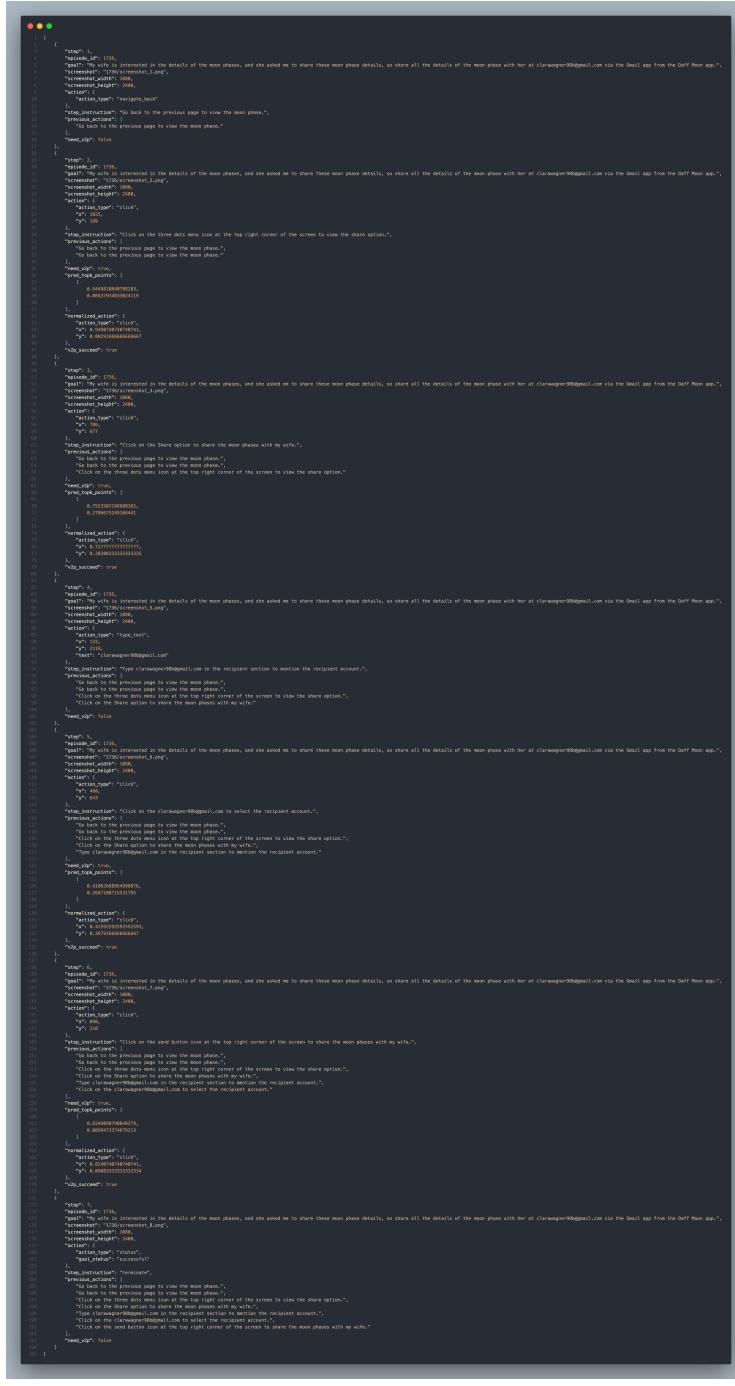


Figure 10: End-to-end real-world application trajectory: "My wife is interested in the details of the moon phases, and she asked me to share these moon phase details, so share all the details of the moon phase with her at clarawagner98b@gmail.com via the Gmail app from the Daff Moon app."

D.2.2 EXECUTION RESULTS

Figure 10 illustrates the complete execution trajectory of the 7-step navigation task. Our V2P-7B model successfully completed the entire trajectory without errors, accurately localizing target elements at each step despite varying interface layouts and contextual changes.

1242 The end-to-end execution demonstrates our V2P-7B model’s robust practical capabilities, achieving
1243 100% task completion rate with consistent localization accuracy across diverse UI elements and
1244 application contexts. With the reliable performance across varying visual conditions and state
1245 transitions. This validation confirms that V2P-7B successfully bridges research benchmarks and real-
1246 world applications, with its powerful grounding capabilities providing stable support and guarantee
1247 for GUI automation.

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295