

GUI-G²: Gaussian Reward Modeling for GUI Grounding

Fei Tang^{1,2*}, Zhangxuan Gu², Zhengxi Lu¹, Xuyang Liu²
 Shuheng Shen^{2†}, Changhua Meng², Wen Wang¹, Wenqi Zhang¹
 Yongliang Shen^{1‡}, Weiming Lu¹, Jun Xiao¹, Yueting Zhuang¹

¹Zhejiang University

²Ant Group

{flysugar, syl}@zju.edu.cn, shuheng.ssh@antgroup.com

Abstract

Graphical User Interface (GUI) grounding maps natural language instructions to precise interface locations for autonomous interaction. Current reinforcement learning approaches use binary rewards that treat elements as hit-or-miss targets, creating sparse signals that ignore the continuous nature of spatial interactions. Motivated by human clicking behavior that naturally forms Gaussian distributions centered on target elements, we introduce GUI Gaussian Grounding Rewards (GUI-G²), a principled reward framework that models GUI elements as continuous Gaussian distributions across the interface plane. GUI-G² incorporates two synergistic mechanisms: Gaussian point rewards model precise localization through exponentially decaying distributions centered on element centroids, while coverage rewards assess spatial alignment by measuring the overlap between predicted Gaussian distributions and target regions. To handle diverse element scales, we develop an adaptive variance mechanism that calibrates reward distributions based on element dimensions. This framework transforms GUI grounding from sparse binary classification to dense continuous optimization, where Gaussian distributions generate rich gradient signals that guide models toward optimal interaction positions. Extensive experiments across ScreenSpot, ScreenSpot-v2, and ScreenSpot-Pro benchmarks demonstrate that GUI-G²-7B substantially outperforms state-of-the-art method UI-TARS-72B, with the most significant improvement of 24.7% on ScreenSpot-Pro. Our analysis reveals that continuous modeling provides superior robustness to interface variations and enhanced generalization to unseen layouts, establishing a new paradigm for spatial reasoning in GUI interaction tasks.

Code — <https://github.com/ZJU-REAL/GUI-G2>

Extended version — <https://arxiv.org/abs/2507.15846>

Introduction

Autonomous GUI agents are revolutionizing human-computer interaction (HCI) by allowing users to control

*This work was done when the first author was an intern at Ant Group.

†Project leader.

‡Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

interfaces with natural language across various applications (Gou et al. 2024; Tang et al. 2025b; Cheng et al. 2024). As the core of these systems, GUI grounding, is the fundamental capability to accurately map natural language instructions to precise pixel coordinates on interface elements (Tang et al. 2025a; Cheng et al. 2024; Lin et al. 2025).

Recent advances in GUI grounding have increasingly adopted reinforcement learning (RL) frameworks (Lu et al. 2025; Luo et al. 2025; Liu et al. 2025b). However, current approaches rely on binary reward systems (Lu et al. 2025; Luo et al. 2025; Yuan et al. 2025; Zhou et al. 2025a) that assign rewards of 1 for coordinates within target bounding boxes and 0 otherwise. This formulation treats GUI interactions as binary hit-or-miss problems, creating sparse learning signals where predictions one pixel outside target regions receive the same zero reward as complete failures. The binary paradigm ignores two critical aspects of interface interaction: first, clicking quality varies continuously with distance from element centers, and second, interface elements are inherently two-dimensional regions with spatial structure, not abstract points. This mismatch between discrete optimization and the continuous geometric nature of GUI interactions severely limits learning efficiency, particularly during early training when models need dense feedback to develop appropriate grounding behaviors.

This discrete approach contradicts empirical evidence from human behavior. Analysis of the AITW dataset (Rawles et al. 2023) reveals that users’ clicks naturally form Gaussian distributions centered on target elements (Figure 1, right), consistent with Fitts’ Law (Fitts 1954; MacKenzie 1992). This pattern demonstrates that spatial targeting inherently follows continuous probability distributions, with click density decreasing smoothly from element centers to edges. Current binary mechanisms completely ignore this fundamental characteristic of HCI.

Building on this insight, we introduce GUI-G² (GUI Gaussian Grounding Rewards), a principled framework that fundamentally reconceptualizes GUI grounding by modeling clicking points as smooth probability distributions across the interface plane. Rather than treating elements as discrete hit-or-miss targets, GUI-G² represents them as continuous Gaussian distributions that provide rich spatial information and dense learning signals. This approach comprises

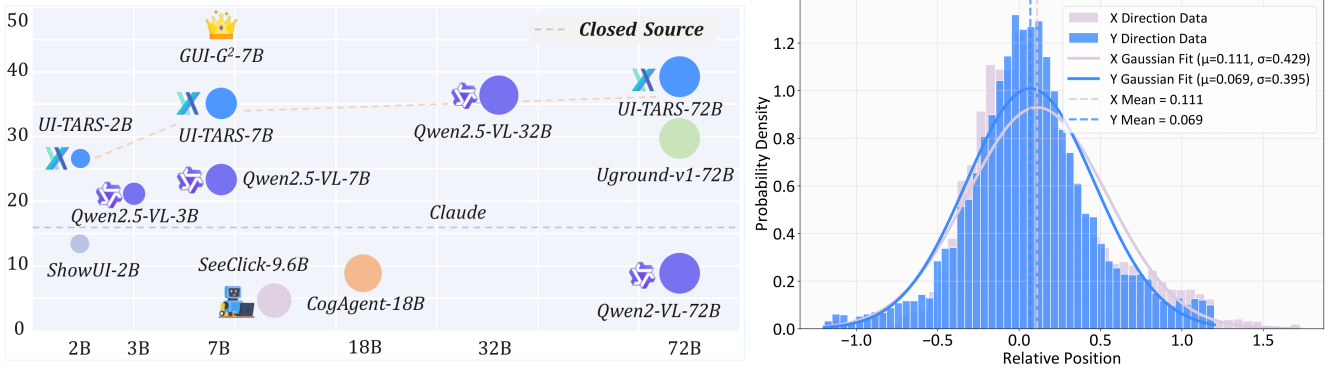


Figure 1: GUI grounding performance and human click behavior. Left: Performance comparison of various models on ScreenSpot-Pro across different parameter scales, where our GUI-G² achieves state-of-the-art performance. Right: Human click distribution from AITW (Rawles et al. 2023) reveals natural Gaussian patterns around target centers ($\mu = 0.111$, $\sigma = 0.429$), validating our design choice of continuous Gaussian rewards over discrete binary feedback.

two complementary mechanisms: First, we design point-based rewards that decrease smoothly with distance from element centers, encouraging precise while maintaining continuous gradients. Second, we introduce coverage-based rewards that measure the spatial overlap between predicted click distributions and target element regions, ensuring comprehensive element targeting.

To accommodate varying element scales, we introduce an adaptive variance mechanism that dynamically adjusts reward distributions according to element dimensions. This ensures consistent learning signals across GUI components while maintaining their distinct geometric properties. GUI-G² transforms GUI grounding from sparse binary optimization to dense continuous reasoning, enabling models to learn fine-grained spatial relationships and develop more robust interaction strategies.

Extensive evaluation on ScreenSpot (Cheng et al. 2024), ScreenSpot-v2 (Wu et al. 2024), and ScreenSpot-Pro (Li et al. 2025) benchmarks demonstrates that our approach achieves substantial improvements over state-of-the-art methods, with accuracy gains up to 4.1%, 3.3%, and 24.7%, respectively. Our analysis reveals superior robustness to interface variations and enhanced generalization to unseen layouts, confirming that continuous spatial modeling provides more fundamental and transferable representations than discrete alternatives. Comprehensive ablation studies validate the synergistic contributions of both Gaussian components and the critical importance of adaptive variance mechanisms for handling interface diversity.

To summarize, our main contributions are three-fold:

- **Continuous Spatial Modeling:** We introduce GUI-G², a principled framework that fundamentally transforms GUI grounding from discrete binary optimization to continuous spatial processes, modeling interface interactions as Gaussian distributions that capture the inherent geometric nature of GUI elements.
- **Dual-Component Reward System:** We propose a novel reward architecture comprising Gaussian point rewards for precise localization and Gaussian coverage rewards

for regional assessment, enhanced with adaptive variance mechanisms that automatically calibrate distributions based on element dimensions.

- **State-of-The-Art Performance:** Extensive experiments demonstrate that GUI-G² achieves substantial improvements with 92.0% accuracy on ScreenSpot, 93.3% on ScreenSpot-v2, and 47.5% on ScreenSpot-Pro, while exhibiting enhanced robustness and generalization compared to discrete reward approaches.

Related Work

GUI Agents. Powered by large language and multi-modal models (Bai et al. 2025; Xu et al. 2025b,a; Zhou et al. 2025b; Gu et al. 2022; Liu et al. 2025a; Xu et al. 2025a), GUI agents are intelligent systems that understand and interact with graphical user interfaces through natural language instructions, enabling automated execution of complex computer tasks (Gou et al. 2024; Zhang et al. 2025a; Tang et al. 2025b; Sun et al. 2025; Shen et al. 2023; Wang et al. 2024c; Wu et al. 2025b,a; Du et al. 2025). These approaches can be broadly categorized into two main paradigms: (1) **Expert Design-Driven Workflow Paradigm:** These approaches typically leverage closed-source multimodal large language models and construct workflows through expertly designed fine-grained modules such as planners (Wang et al. 2024b; Zhang et al. 2024) and grounders (Gou et al. 2024; Lin et al. 2025). The Mobile-Agent series (Wang et al. 2025, 2024a,b), AppAgent series (Zhang et al. 2023; Li et al. 2024), and UFO series (Zhang et al. 2024, 2025b) accomplish various tasks through these workflow-based approaches. These GUI agents typically consist of planners and grounders, where planners employ closed-source large language models such as GPT-4o (OpenAI 2024) and Claude (Anthropic 2024) for task planning. For grounding components, there are two main approaches: one utilizes HTML and DOM tree structures for screen understanding (Rawles et al. 2023; Zhang et al. 2023), while the other employs visual tools such as OCR (Du et al. 2020), SAM (Kirillov et al. 2023) for element localiza-

tion. However, this reliance on pre-programmed workflows makes frameworks inherently non-scalable, consuming substantial manual effort and proving difficult to extend to new domains (Qin et al. 2025). **(2) Data-Driven Training Paradigm:** These approaches employ specialized MLLMs trained specifically for GUI understanding and interaction through data-driven methodologies (Qin et al. 2025; Gou et al. 2024; Tang et al. 2025a; Wu et al. 2024; Wang et al. 2024c; Tang et al. 2025a). These works achieve GUI-specific capabilities by collecting large-scale GUI corpora for fine-tuning to develop models tailored for GUI tasks. For example, UI-TARS (Qin et al. 2025) develops an end-to-end native GUI agent through large-scale GUI screenshots for enhanced perception and action traces for unified action modeling across platforms. However, due to supervised fine-tuning limitations (Chu et al. 2025), these methods still face generalization challenges when encountering novel interface scenarios (Lu et al. 2025).

Reinforcement Fine-Tuning (RFT). Since the release of DeepSeek-R1 (DeepSeek-AI 2025), rule-based reward has been applied across various domains (Dai et al. 2025; Shen et al. 2025). Researchers have begun applying this approach to GUI tasks (Gu et al. 2025). GUI-R1 (Luo et al. 2025) and UI-R1 (Lu et al. 2025) apply verifiable reward paradigms to GUI tasks, representing pioneering efforts in this direction while demonstrating the potential of RFT. InfGUI-R1 (Liu et al. 2025b) similarly follows the R1 paradigm, employing two-stage training to inject reasoning capabilities into the model. GUI-G1 (Zhou et al. 2025a) reanalyzes existing problems in current R1-based GUI agents and designs controllable box size rewards for GUI grounding tasks. However, most previous methods rely on binary reward that focus on hit-or-miss feedback, creating sparse signals that struggle to provide effective guidance during early training stages. Recent works like SE-GUI (Yuan et al. 2025) and LPO (Tang et al. 2025c) propose distance-based continuous rewards that decay with Euclidean distance from target coordinates. While these approaches provide smoother signals than binary alternatives, they treat GUI elements as point targets, missing the inherent two-dimensional spatial structure of interface interactions. We address this limitation by proposing a principled Gaussian continuous reward mechanism that models GUI elements as 2D probability distributions, providing geometrically-aware feedback that captures both precise localization and spatial coverage.

Methodology

We introduce GUI-G² (GUI Gaussian Grounding Rewards), a principled framework that reformulates GUI grounding rewards from discrete binary signals to continuous Gaussian distributions. As illustrated in Figure 2, our approach comprises three key innovations: (1) Gaussian point rewards that model localization precision, (2) Gaussian coverage rewards that capture spatial overlap, and (3) adaptive variance mechanisms that scale with element dimensions. This continuous formulation addresses binary rewards’ fundamental limitation by providing learning signals for near-misses through smooth gradients throughout the spatial domain.

Problem Formulation

GUI grounding maps natural language instructions to pixel-level targets on graphical interfaces. Given a screenshot s and instruction i , the model must predict a bounding box $\mathbf{b}^p = [x_1^p, y_1^p, x_2^p, y_2^p]$ that localizes the element described by i , where (x_1, y_1) and (x_2, y_2) denote the top-left and bottom-right corners respectively. The ground truth is annotated as $\mathbf{b}^{gt} = [x_1^{gt}, y_1^{gt}, x_2^{gt}, y_2^{gt}]$.

In the RL formulation, the model generates a sequence of tokens representing the predicted bounding box coordinates. The standard evaluation criterion checks whether the predicted center $(c_x^p, c_y^p) = (\frac{x_1^p+x_2^p}{2}, \frac{y_1^p+y_2^p}{2})$ falls within \mathbf{b}^{gt} . Our reward function $R(\mathbf{b}^p, \mathbf{b}^{gt})$ transforms this discrete success metric into continuous spatial feedback. Unlike binary rewards that provide no gradient for near-misses, GUI-G² generates dense learning signals that vary smoothly with prediction quality, enabling more efficient policy optimization through richer supervision.

Gaussian Reward Modeling

We model GUI elements as 2D Gaussian distributions to capture the continuous nature of spatial interactions. This approach transforms discrete bounding boxes into smooth probability distributions that naturally encode spatial uncertainty and provide rich gradient information.

Gaussian Representation. For each GUI element with bounding box $\mathbf{b} = [x_1, y_1, x_2, y_2]$, we construct a 2D Gaussian distribution:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (1)$$

where $\mathbf{x} = (x, y)$ represents a position in the 2D interface space, $\boldsymbol{\mu} = (c_x, c_y) = (\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$ is the element’s geometric center, and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$ is a diagonal covariance matrix. The diagonal structure assumes independence between x and y dimensions, simplifying computation while maintaining expressiveness.

Gaussian Point Rewards. The point reward evaluates localization precision by measuring how well the predicted center aligns with the target element’s Gaussian distribution. Given a predicted bounding box with center $\boldsymbol{\mu}_p = (c_x^p, c_y^p)$ and ground truth center $\boldsymbol{\mu}_{gt} = (c_x^{gt}, c_y^{gt})$, we compute:

$$R_{point} = \mathcal{N}(\boldsymbol{\mu}_p; \boldsymbol{\mu}_{gt}, \boldsymbol{\Sigma}_{gt}) = \exp\left(-\frac{1}{2}\left[\frac{(c_x^p - c_x^{gt})^2}{\sigma_x^{gt^2}} + \frac{(c_y^p - c_y^{gt})^2}{\sigma_y^{gt^2}}\right]\right) \quad (2)$$

This provides several key properties. First, it decreases smoothly and exponentially with distance, ensuring continuous gradients throughout the spatial domain. Second, the rate of decay is controlled by the variance parameters, allowing flexible adaptation to different element characteristics.

Gaussian Coverage Rewards. While point rewards optimize center alignment, GUI interactions often succeed when

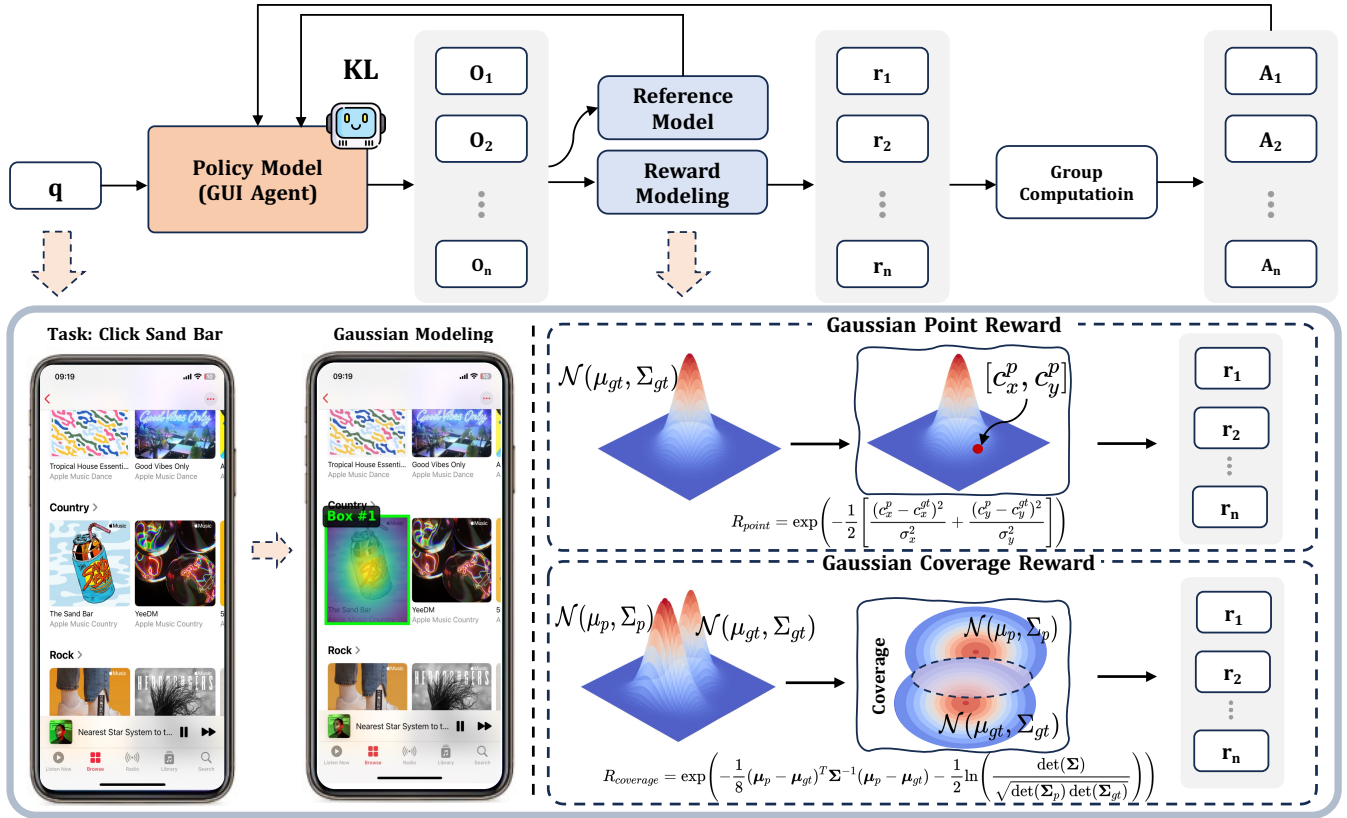


Figure 2: GUI Gaussian Grounding Rewards (GUI-G²). Our framework transforms GUI grounding through continuous Gaussian modeling. Given a task instruction and screenshot, the policy model generates multiple predictions that are evaluated using our dual reward mechanism. Gaussian Point Rewards assess localization precision while Gaussian Coverage Rewards measure spatial overlap, together providing dense learning signals that guide policy optimization.

clicking anywhere within element boundaries. Coverage rewards capture this regional aspect by measuring overlap between predicted and target Gaussian distributions. We quantify this using the Bhattacharyya coefficient (BC):

$$BC(\mathcal{N}_p, \mathcal{N}_{gt}) = \int \sqrt{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{gt}, \boldsymbol{\Sigma}_{gt})} d\mathbf{x} \quad (3)$$

For Gaussian distributions, this integral has a closed-form solution, formulated as:

$$R_{coverage} = \exp\left(-\frac{1}{8}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_{gt})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_{gt}) - \frac{1}{2} \ln\left(\frac{\det(\boldsymbol{\Sigma})}{\sqrt{\det(\boldsymbol{\Sigma}_p) \det(\boldsymbol{\Sigma}_{gt})}}\right)\right) \quad (4)$$

where $\boldsymbol{\Sigma} = \frac{\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_{gt}}{2}$ is the average covariance. The first term penalizes center misalignment weighted by the combined uncertainty, while the second term measures size and shape similarity between distributions.

Adaptive Variance Mechanism. GUI elements span diverse scales, from tiny icons to full-screen panels. Fixed variance parameters would either over-constrain large ele-

ments or under-constrain small ones. We introduce an adaptive mechanism scaling variance with element dimensions:

$$\sigma_x = \alpha \cdot (x_2 - x_1), \quad \sigma_y = \alpha \cdot (y_2 - y_1) \quad (5)$$

where α is a scaling factor controlling the relative influence of element size on the standard deviations. The intuition is straightforward: larger elements naturally tolerate greater spatial uncertainty in interactions. A small icon requires precise targeting, while a large button can be activated across a wider region. By making the Gaussian spread proportional to element size, we ensure the reward function respects this natural interaction pattern. This adaptive mechanism applies to both point and coverage rewards.

Reinforcement Learning with GUI-G²

We combine both reward components to leverage precise localization and spatial coverage:

$$R_{total} = \nu \cdot R_{point} + \gamma \cdot R_{coverage} \quad (6)$$

where ν and γ balance the contribution of each component. The point reward drives the model toward accurate center positioning, while the coverage reward ensures appropriate spatial extent. This dual objective mirrors human interaction patterns: users aim for element centers but can successfully

interact anywhere within boundaries.

We integrate GUI-G² into Group Relative Policy Optimization (GRPO) (Shao et al. 2024), which estimates advantages using multiple sampled responses. For each instruction, we sample N predictions and compute their rewards under GUI-G². The advantage for response i is:

$$A_i = \frac{R_{total}(\tau_i) - \text{mean}(\{R_{total}(\tau_j)\}_{j=1}^N)}{\text{std}(\{R_{total}(\tau_j)\}_{j=1}^N)} \quad (7)$$

This normalization stabilizes gradients across element types and sizes. The policy optimization objective becomes:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta_{old}}} \left[\sum_t \min \left(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) A_t \right) - \beta \mathbb{D}_{KL}[\pi_{\theta} \parallel \pi_{ref}] \right] \quad (8)$$

where $r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)}$ is the probability ratio, ϵ controls the trust region, and β weights the KL regularization. The continuous nature of GUI-G² rewards fundamentally transforms the optimization landscape. While binary rewards create a discontinuous surface with sharp cliffs at bounding box edges, our Gaussian formulation produces smooth gradients everywhere in the spatial domain. This smoothness is crucial during early training: even when predictions are far from targets, the exponentially decaying Gaussian signals provide clear directional guidance toward improvement.

Experiments

Experiment Setup

Implementation Details. We implement our GUI-G² using Qwen2.5-VL-7B-Instruct (Bai et al. 2025) within the VLM-R1 framework (Shen et al. 2025). Training is conducted on 8 A100-80G GPUs for one epoch with learning rate 1e-6, batch size 8, $\beta = 0.04$, and $\alpha = 0.5$. We freeze the vision module and use Flash Attention 2 (Dao 2023) with bfloat16 precision. Unless otherwise specified, ν and γ are set to 1.0.

Training Dataset and Evaluation Benchmarks. Our training data comprises approximately 100K GUI grounding instances sampled from Widget Captioning (Cheng et al. 2024), UI RefExp (Bai et al. 2021), ShowUI-web (Lin et al. 2025), and OmniAct (Kapoor et al. 2024), covering diverse interface types across mobile, desktop, and web platforms. We evaluate on three benchmarks: ScreenSpot (Cheng et al. 2024) (SS-v1) and ScreenSpot-v2 (Wu et al. 2024) (SS-v2) for general GUI grounding, and ScreenSpot-Pro (Li et al. 2025) (SS-Pro) for high-resolution professional software interfaces. Following standard protocol (Cheng et al. 2024; Lin et al. 2025), predictions are considered correct when the predicted center falls within the ground truth bounding box.

Main Results

We evaluate GUI-G²-7B on three benchmarks: SS-v1, SS-v2, and SS-Pro. Table 1 shows GUI-G² achieves **state-of-the-art performance** among all RL and SFT approaches,

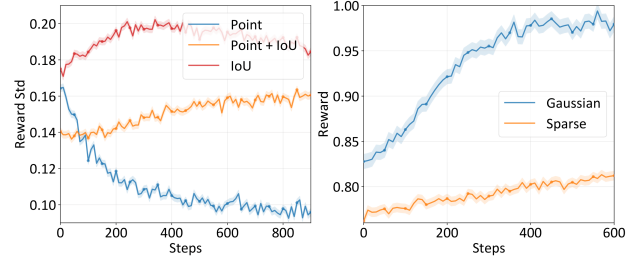


Figure 3: Sparse reward training dynamics. Training dynamics of sparse reward variants (Point, IoU, Point+IoU) showing reward standard deviation and convergence.

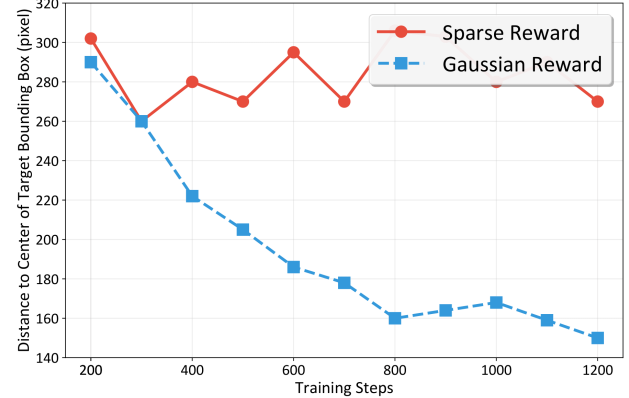


Figure 4: Distance convergence comparison. Distance to target center over training steps, where Gaussian Reward demonstrates monotonic convergence while Sparse Reward exhibits erratic fluctuations.

reaching 92.0% on SS-v1, 93.3% on SS-v2, and 47.5% on SS-Pro. On SS-Pro, GUI-G² surpasses UI-TARS-72B by **10.8%** while using only **10%** of its parameters, demonstrating that continuous Gaussian rewards enable efficient optimization in small models. Compared to other continuous reward methods (LPO-8B: 90.5%, SE-GUI-7B: 90.3% on SS-v2), GUI-G² shows clear advantages. While these methods use distance-based rewards treating elements as points, our dual Gaussian formulation models both precise localization (point rewards) and spatial extent (coverage rewards), capturing the complete interaction space. On SS-Pro’s text elements, we achieve 64.7% versus UI-TARS-72B’s 50.9%, indicating Gaussian rewards benefit fine-grained spatial tasks. These results establish continuous Gaussian modeling as an effective foundation for GUI grounding.

Reward Design Analysis

Binary vs. Continuous Rewards. We investigate binary versus continuous rewards by implementing three sparse baselines: Point rewards (activating when predicted centers fall within target boxes), IoU rewards (triggering when overlap exceeds 0.5), and their combination. We select 10 challenging SS-v2 samples with incorrect initial predictions, tracking average distance from predicted to ground truth

Model	CAD		Dev		Creative		Scientific		Office		OS		Avg.				
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	SS-Pro	SS-v1	SS-v2
<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	18.8	20.1
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	83.0	-
<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	55.5	80.9
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	84.7	88.8
<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	53.4	55.1
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	47.4	-
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	82.5	84.1
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	75.1	77.3
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-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	89.5	91.6
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	88.4	90.3
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	-	88.6
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	-	91.7
GUI-Actor-7B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44.6	88.3	92.1
<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	83.3	85.4
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	89.2	89.5
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	-	-	-	-	-
InfGUI-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	87.5	-
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	-	90.3
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	88.2	90.3
GUI-G ² -3B	38.1	15.6	60.3	12.4	52.5	9.1	63.1	20.0	63.0	19.2	41.1	6.7	53.0	13.1	37.8	90.8	91.0
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	92.0	93.3

Table 1: Comparison of different agent models across various task categories based on Text, Icon, and Average scores on SS-Pro. “-” indicates unreported results in original papers.

centers across 8 sampled responses every 200 steps. Figure 3 reveals critical limitations of sparse signals. Binary rewards generate erratic optimization with severe fluctuations in reward values and spatial convergence. Point rewards plateau early, while IoU demonstrates poor learning dynamics. Sparse methods show no consistent distance reduction, oscillating between 200-400 pixels without progress (Figure 4). In contrast, GUI-G² exhibits smooth monotonic convergence from 290px to 150px. Table 3 quantifies this: GUI-G² achieves 93.3% accuracy, surpassing the best sparse baseline (Point: 87.4%) by 5.9%. This gap emerges because Gaussian rewards provide informative gradients at every spatial position, enabling learning from any distance. Binary rewards create discrete cliffs at box edges where gradients vanish, leaving no guidance for near-miss predictions. Our continuous formulation creates smooth paths toward targets from any starting position.

Inside vs. Outside Boundary Rewards: Why Continuous Everywhere Matters. A natural question arises: should re-

wards be provided only within target boundaries or everywhere in the interface? We implement an Inside Gaussian (IG) baseline that applies our Gaussian formulation only when predictions fall within ground truth boxes, reverting to zero otherwise. Figure 5 shows GUI-G² outperforms IG by **4.9%**, despite using identical Gaussian formulations. This reveals that restricting rewards to successful predictions, even with continuous formulations, recreates the fundamental problem of sparse signals. By providing Gaussian feedback throughout the entire interface plane, GUI-G² enables models to learn from every prediction, creating smooth optimization paths from any starting position to target elements.

Dual Gaussian Components: Point Precision and Spatial Coverage. GUI-G²’s dual formulation addresses complementary aspects of interface interaction. Ablation studies in Figure 5 demonstrate that removing either component significantly degrades performance: 92.1% without coverage rewards and 90.2% without point rewards, compared to 93.3% with both. Point rewards alone optimize for pre-

Model	ScreenSpot Acc(%)				ScreenSpot-v2 Acc (%)			
	Mobile		Desktop		Mobile		Desktop	
SE-GUI-7B	85.6	91.4	86.5	88.2	95.2	87.1	87.0	90.3
GUI-G ² -7B	94.0	92.8	89.0	92.0	95.6	92.8	91.1	93.3

Table 2: Gaussian vs. Distance-based Rewards. GUI-G² outperforms SE-GUI-7B on SS-v1 and SS-v2, validating the superiority of our Gaussian reward design.

Reward Type	Mobile		Desktop		Web		Avg
	Text	Icon	Text	Icon	Text	Icon	
<i>Sparse Reward</i>							
Point	97.9	87.2	88.7	72.1	84.9	79.8	87.4
IoU	95.9	86.7	87.1	69.3	88.4	77.3	85.8
Point + IoU	97.2	86.7	88.1	68.6	88.9	78.8	86.5
<i>Dense Reward</i>							
GUI-G ² -7B	98.3	91.9	95.4	89.3	94.0	87.7	93.3

Table 3: Results of sparse and dense methods on SS-v2. GUI-G² outperforms sparse reward baselines.

cise center localization but ignore that users can successfully click anywhere within element boundaries. Coverage rewards alone measure spatial overlap but lack the precision to guide models toward optimal clicking positions. The improvement from their combination confirms that effective GUI grounding requires modeling both aspects simultaneously, reflecting how humans naturally aim for element centers while accepting clicks anywhere within boundaries.

GUI-G² vs. Distance-Based Rewards. SE-GUI-7B represents an alternative continuous approach using normalized Euclidean distance. Table 2 shows GUI-G² consistently outperforms SE-GUI by 3.8% on SS-v1, 3.0% on SS-v2 and 0.2% on SS-Pro. This gap highlights two fundamental differences. First, SE-GUI treats elements as point targets, computing distances to centers without considering spatial extent. Second, it applies different formulas inside versus outside bounding boxes, creating gradient discontinuities at boundaries. Our unified Gaussian formulation provides smooth gradients everywhere while explicitly modeling both localization precision and spatial coverage, better capturing the continuous nature of GUI interactions.

Adaptive Variance Mechanism. GUI elements span diverse scales from tiny icons to full-screen panels, and human clicking tolerance varies with element dimensions: larger elements naturally accommodate greater spatial uncertainty, while small icons require precise targeting. To handle this diversity, we propose an adaptive variance mechanism that scales reward distributions based on element size. We implement: (i) 1 σ Principle: $\sigma_x = \text{width}/2$, $\sigma_y = \text{height}/2$; (ii) 2 σ Principle: $\sigma_x = \text{width} \times 2$, $\sigma_y = \text{height} \times 2$; (iii) 3 σ Principle: $\sigma_x = \text{width} \times 3$, $\sigma_y = \text{height} \times 3$; and (iv) w/o adaptive σ : fixed variance for all elements. Figure 6 shows our adaptive mechanism with $\alpha = 0.5$ achieves peak performance at 93.3%, substantially outperforming fixed variance

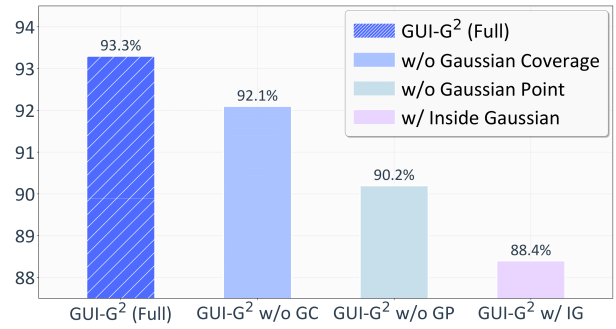


Figure 5: Ablation of Gaussian Component. Both components contribute to the final performance of 93.3% on SS-v2.

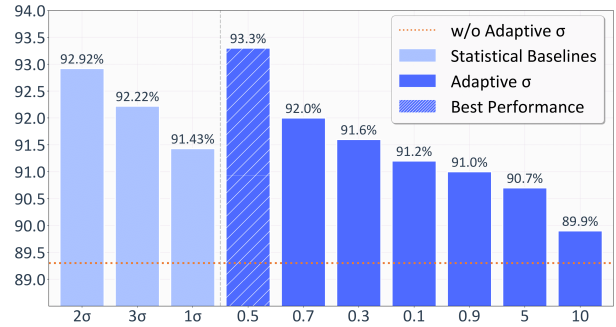


Figure 6: Hyper-parameter sensitivity analysis of σ . Performance peaks at $\alpha = 0.5$ with 93.3.0% accuracy on SS-v2.

approaches (87.8%) by +5.5%. This optimal value aligns with the 2 σ statistical principle (92.92%), demonstrating that effective GUI grounding emerges from balanced spatial tolerance. The 1 σ principle (91.43%) proves overly restrictive, while the 3 σ principle (92.22%) shows excessive tolerance dilutes precision. Our adaptive mechanism significantly outperforms the optimal 2 σ baseline, demonstrating that element-specific calibration optimally balances spatial flexibility and targeting precision.

Conclusion

We propose GUI-G², a reward modeling framework that reconceptualizes GUI grounding as a continuous spatial optimization task. Unlike approaches relying on sparse binary rewards, GUI-G² leverages Gaussian point rewards and coverage rewards to provide dense, geometrically-aware feedback. By modeling GUI elements as 2D Gaussian distributions with adaptive variance, our method captures both localization precision and spatial coverage, enabling more efficient learning and better generalization. Evaluated on SS-v1, SS-v2, and SS-Pro, GUI-G²-7B outperforms state-of-the-art models, achieving up to 24.7% improvement over UI-TARS-72B on high-resolution professional interfaces with 90% fewer parameters.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62506332), National Key Research and Development Project (No. 2024YFB3312900), Ant Group Research Intern Program and Ant Group through CCF-Ant Research Fund.

References

- Anthropic. 2024. Claude Computer Use. Available at: <https://www.anthropic.com/news/developing-computer-use>.
- Bai, C.; Zang, X.; Xu, Y.; Sunkara, S.; Rastogi, A.; Chen, J.; and y Arcas, B. A. 2021. UIBert: Learning Generic Multimodal Representations for UI Understanding. arXiv:2107.13731.
- Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923.
- Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents. arXiv:2401.10935.
- Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. arXiv:2501.17161.
- Dai, Y.; Yang, S.; Wang, G.; Deng, Y.; Zhang, Z.; Yin, J.; Zeng, P.; Ying, Z.; Meng, C.; Yi, C.; Zhou, Y.; Wang, W.; and Lu, S. 2025. Careful Queries, Credible Results: Teaching RAG Models Advanced Web Search Tools with Reinforcement Learning.
- Dao, T. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691.
- DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.
- Du, Y.; Li, C.; Guo, R.; Yin, X.; Liu, W.; Zhou, J.; Bai, Y.; Yu, Z.; Yang, Y.; Dang, Q.; and Wang, H. 2020. PP-OCR: A Practical Ultra Lightweight OCR System. arXiv:2009.09941.
- Du, Y.; Yan, Y.; Tang, F.; Lu, Z.; Zong, C.; Lu, W.; Jiang, S.; and Shen, Y. 2025. Test-Time Reinforcement Learning for GUI Grounding via Region Consistency.
- Fitts, P. M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 74: 381–391.
- Gou, B.; Wang, R.; Zheng, B.; Xie, Y.; Chang, C.; Shu, Y.; Sun, H.; and Su, Y. 2024. Navigating the Digital World as Humans Do: Universal Visual Grounding for GUI Agents. arXiv:2410.05243.
- Gu, Z.; Meng, C.; Wang, K.; Lan, J.; Wang, W.; Gu, M.; and Zhang, L. 2022. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4583–4592.
- Gu, Z.; Zeng, Z.; Xu, Z.; Zhou, X.; Shen, S.; Liu, Y.; Zhou, B.; Meng, C.; Xia, T.; Chen, W.; Wen, Y.; Dou, J.; Tang, F.; Lin, J.; Liu, Y.; Guo, Z.; Gong, Y.; Jia, H.; Gao, C.; Guo, Y.; Deng, Y.; Guo, Z.; Chen, L.; and Wang, W. 2025. UI-Venus Technical Report: Building High-performance UI Agents with RFT.
- Kapoor, R.; Butala, Y. P.; Russak, M.; Koh, J. Y.; Kamble, K.; Alshikh, W.; and Salakhutdinov, R. 2024. OmniACT: A Dataset and Benchmark for Enabling Multimodal Generalist Autonomous Agents for Desktop and Web.
- Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. arXiv:2304.02643.
- Li, K.; Meng, Z.; Lin, H.; Luo, Z.; Tian, Y.; Ma, J.; Huang, Z.; and Chua, T.-S. 2025. ScreenSpot-Pro: GUI Grounding for Professional High-Resolution Computer Use.
- Li, Y.; Zhang, C.; Yang, W.; Fu, B.; Cheng, P.; Chen, X.; Chen, L.; and Wei, Y. 2024. AppAgent v2: Advanced Agent for Flexible Mobile Interactions. arXiv:2408.11824.
- Lin, K. Q.; Li, L.; Gao, D.; Yang, Z.; Wu, S.; Bai, Z.; Lei, W.; Wang, L.; and Shou, M. Z. 2025. ShowUI: One Vision-Language-Action Model for GUI Visual Agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Liu, X.; Gui, X.; Zhang, Y.; and Zhang, L. 2025a. Mixing Importance with Diversity: Joint Optimization for KV Cache Compression in Large Vision-Language Models. arXiv preprint arXiv:2510.20707.
- Liu, Y.; Li, P.; Xie, C.; Hu, X.; Han, X.; Zhang, S.; Yang, H.; and Wu, F. 2025b. InfiGUI-R1: Advancing Multimodal GUI Agents from Reactive Actors to Deliberative Reasoners.
- Lu, Z.; Chai, Y.; Guo, Y.; Yin, X.; Liu, L.; Wang, H.; Xiao, H.; Ren, S.; Xiong, G.; and Li, H. 2025. UI-R1: Enhancing Efficient Action Prediction of GUI Agents by Reinforcement Learning.
- Luo, R.; Wang, L.; He, W.; and Xia, X. 2025. GUI-R1 : A Generalist R1-Style Vision-Language Action Model For GUI Agents.
- MacKenzie, I. S. 1992. Fitts’ Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction*, 7(1): 91–139.
- OpenAI. 2024. Introducing GPT-4o. Available at: <https://openai.com/index/hello-gpt-4o>.
- Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; Zhong, W.; Li, K.; Yang, J.; Miao, Y.; Lin, W.; Liu, L.; Jiang, X.; Ma, Q.; Li, J.; Xiao, X.; Cai, K.; Li, C.; Zheng, Y.; Jin, C.; Li, C.; Zhou, X.; Wang, M.; Chen, H.; Li, Z.; Yang, H.; Liu, H.; Lin, F.; Peng, T.; Liu, X.; and Shi, G. 2025. UI-TARS: Pioneering Automated GUI Interaction with Native Agents. arXiv:2501.12326.
- Rawles, C.; Li, A.; Rodriguez, D.; Riva, O.; and Lillicrap, T. 2023. Android in the Wild: A Large-Scale Dataset for Android Device Control.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv:2402.03300*.

Shen, H.; Liu, P.; Li, J.; Fang, C.; Ma, Y.; Liao, J.; Shen, Q.; Zhang, Z.; Zhao, K.; Zhang, Q.; Xu, R.; and Zhao, T. 2025. VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model. *arXiv:2504.07615*.

Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; and Zhuang, Y. 2023. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. *arXiv:2303.17580*.

Sun, Q.; Cheng, K.; Ding, Z.; Jin, C.; Wang, Y.; Xu, F.; Wu, Z.; Jia, C.; Chen, L.; Liu, Z.; Kao, B.; Li, G.; He, J.; Qiao, Y.; and Wu, Z. 2025. OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis. *arXiv:2412.19723*.

Tang, F.; Shen, Y.; Zhang, H.; Chen, S.; Hou, G.; Zhang, W.; Zhang, W.; Song, K.; Lu, W.; and Zhuang, Y. 2025a. Think Twice, Click Once: Enhancing GUI Grounding via Fast and Slow Systems.

Tang, F.; Xu, H.; Zhang, H.; Chen, S.; Wu, X.; Shen, Y.; Zhang, W.; Hou, G.; Tan, Z.; Yan, Y.; Song, K.; Shao, J.; Lu, W.; Xiao, J.; and Zhuang, Y. 2025b. A Survey on (M)LLM-Based GUI Agents.

Tang, J.; Xia, Y.; Wu, Y.-F.; Hu, Y.; Chen, Y.; Chen, Q.-G.; Xu, X.; Wu, X.; Lu, H.; Ma, Y.; Lu, S.; and Chen, Q. 2025c. LPO: Towards Accurate GUI Agent Interaction via Location Preference Optimization. *arXiv:2506.09373*.

Wang, J.; Xu, H.; Jia, H.; Zhang, X.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024a. Mobile-Agent-v2: Mobile Device Operation Assistant with Effective Navigation via Multi-Agent Collaboration. *arXiv:2406.01014*.

Wang, J.; Xu, H.; Ye, J.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024b. Mobile-Agent: Autonomous Multi-Modal Mobile Device Agent with Visual Perception. *arXiv:2401.16158*.

Wang, K.; Xia, T.; Gu, Z.; Zhao, Y.; Shen, S.; Meng, C.; Wang, W.; and Xu, K. 2024c. E-ant: A large-scale dataset for efficient automatic gui navigation. *arXiv preprint arXiv:2406.14250*.

Wang, Z.; Xu, H.; Wang, J.; Zhang, X.; Yan, M.; Zhang, J.; Huang, F.; and Ji, H. 2025. Mobile-Agent-E: Self-Evolving Mobile Assistant for Complex Tasks. *arXiv:2501.11733*.

Wu, Z.; Cheng, P.; Wu, Z.; Dong, L.; and Zhang, Z. 2025a. GEM: Gaussian Embedding Modeling for Out-of-Distribution Detection in GUI Agents.

Wu, Z.; Huang, H.; Lou, X.; Qu, X.; Cheng, P.; Wu, Z.; Liu, W.; Zhang, W.; Wang, J.; Wang, Z.; and Zhang, Z. 2025b. VeriOS: Query-Driven Proactive Human-Agent-GUI Interaction for Trustworthy OS Agents.

Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P. P.; and Qiao, Y. 2024. OS-ATLAS: A Foundation Action Model for Generalist GUI Agents. *arXiv:2410.23218*.

Xu, H.; Mei, X.; Yan, Y.; Zhou, R.; Zhang, W.; Lu, W.; Zhuang, Y.; and Shen, Y. 2025a. Easysteer: A unified framework for high-performance and extensible llm steering. *arXiv preprint arXiv:2509.25175*.

Xu, H.; Yan, Y.; Shen, Y.; Zhang, W.; Hou, G.; Jiang, S.; Song, K.; Lu, W.; Xiao, J.; and Zhuang, Y. 2025b. Mind the Gap: Bridging Thought Leap for Improved Chain-of-Thought Tuning. *arXiv preprint arXiv:2505.14684*.

Yuan, X.; Zhang, J.; Li, K.; Cai, Z.; Yao, L.; Chen, J.; Wang, E.; Hou, Q.; Chen, J.; Jiang, P.-T.; and Li, B. 2025. Enhancing Visual Grounding for GUI Agents via Self-Evolutionary Reinforcement Learning.

Zhang, C.; He, S.; Qian, J.; Li, B.; Li, L.; Qin, S.; Kang, Y.; Ma, M.; Liu, G.; Lin, Q.; Rajmohan, S.; Zhang, D.; and Zhang, Q. 2025a. Large Language Model-Brained GUI Agents: A Survey.

Zhang, C.; Huang, H.; Ni, C.; Mu, J.; Qin, S.; He, S.; Wang, L.; Yang, F.; Zhao, P.; Du, C.; Li, L.; Kang, Y.; Jiang, Z.; Zheng, S.; Wang, R.; Qian, J.; Ma, M.; Lou, J.-G.; Lin, Q.; Rajmohan, S.; and Zhang, D. 2025b. UFO2: The Desktop AgentOS.

Zhang, C.; Li, L.; He, S.; Zhang, X.; Qiao, B.; Qin, S.; Ma, M.; Kang, Y.; Lin, Q.; Rajmohan, S.; Zhang, D.; and Zhang, Q. 2024. UFO: A UI-Focused Agent for Windows OS Interaction. *arXiv:2402.07939*.

Zhang, C.; Yang, Z.; Liu, J.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2023. AppAgent: Multimodal Agents as Smartphone Users. *arXiv:2312.13771*.

Zhou, Y.; Dai, S.; Wang, S.; Zhou, K.; Jia, Q.; and Xu, J. 2025a. GUI-G1: Understanding R1-Zero-Like Training for Visual Grounding in GUI Agents.

Zhou, Y.; Tang, J.; Yang, S.; Xiao, X.; Dai, Y.; Yang, W.; Gou, C.; Xia, X.; and Chua, T.-S. 2025b. Logic Unseen: Revealing the Logical Blindspots of Vision-Language Models. *arXiv preprint arXiv:2508.11317*.