Learning from Reliable Negatives: Confidence-Anchored Test-Time Adaptation for GUI Grounding

Yizhou Liu*[1], Fei Tang*[1], Yuchen Yan[1], Zhengxi Lu[1], Songqin Nong[2], Tao Jiang[2], Wenhao Xu[2], Wenqi Zhang[1], Weiming Lu[1], Jun Xiao[1], and Yongliang Shen†[1]

[1] Zhejiang University
[2] Ant Group
{22551294,syl}@zju.edu.cn

Abstract. Graphical User Interface (GUI) grounding is essential for autonomous agents to map natural language instructions to precise screen coordinates. However, existing supervised fine-tuning and reinforcement learning methods are constrained by the high cost of annotation, creating a scalability bottleneck. In this paper, we introduce a label-free test-time training paradigm driven by two key insights: (1) confidence patterns in coordinate tokens are a better indicator than full-sequence confidence, and (2) in sparse GUI coordinate spaces, negative samples offer more reliable learning signals than potentially noisy positive ones. We first propose **C**onfidence-**A**nchored **L**earning (CAL), which utilizes coordinate-token confidence to filter pseudo-labels and assign distance-based binary rewards. Building on this, we develop **C**onfidence-**A**nchored **N**egative **L**earning (CANL), which exclusively optimizes the model using negative samples to bypass the risks of incorrect positive samples. Experimental results demonstrate that CANL-7B achieves 92.1% on ScreenSpot-V2. On more challenging ScreenSpot-Pro, CANL-7B reaches 33.8%, an 8.9% absolute improvement over the base model. Our findings establish coordinate-token confidence as a powerful alternative to manual annotations for scalable GUI agent development.

Keywords: GUI Grounding · Test-Time Training

1 Introduction

GUI agents have emerged as a critical technology for automating human-computer interaction, enabling natural language commands to be translated into precise interface actions. While early rule-based systems required extensive manual engineering and lacked adaptability [41,58], the advent of multimodal large language models (MLLMs) [2–4, 28, 55, 59] has opened new possibilities for flexible GUI understanding and interaction [11, 13, 17, 37, 39, 47, 48, 50, 54, 57]. At the heart of

* Equal contribution, † Corresponding author.

these agents lies GUI grounding, the task of mapping natural language instructions to specific interface elements through coordinate prediction [7, 21, 45].

Current approaches to GUI grounding employ two main training paradigms, as illustrated in Fig. 1: (1) Supervised Fine-Tuning (SFT) [7, 12, 15, 21, 29, 33, 38, 44, 45, 49], which directly learns from ground-truth bounding box annotations, and (2) Reinforcement Learning with Verifiable Rewards (RLVR) [6,18,22–26,40,53,56,60], which requires ground-truth labels to compute accurate rewards for policy optimization. However, both paradigms face a fundamental challenge: the heavy reliance on annotated data. Obtaining pixel-level bounding box annotations is prohibitively expensive and time-consuming, requiring manual labeling of precise coordinates for each UI element. This dependency on labeled data severely limits the scalability and practical deployment of GUI agents across diverse applications and domains.

A natural question arises: can we enhance GUI grounding capabilities without explicit supervision? Test-time reinforcement learning (TTRL) [62] offers a promising direction, having demonstrated success in mathematical reasoning tasks through label-free adaptation. However, applying TTRL to GUI grounding presents unique challenges. The region consistency supervision based on predicted bounding boxes introduces large supervision errors, leading to marginal performance gains [9]. Unlike mathematical problems where correctness can be verified through symbolic computation, GUI tasks require spatial reasoning over visual elements where ground truth is unavailable during inference.

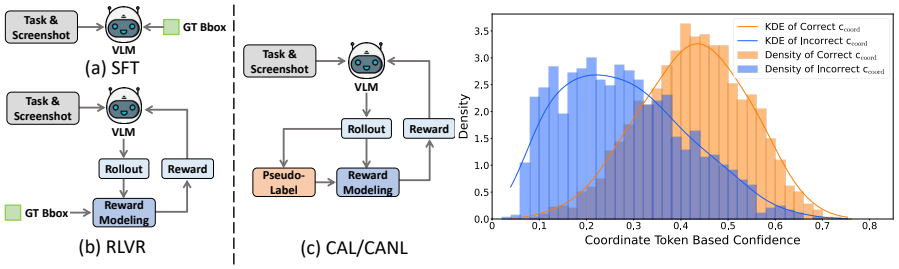

Fig. 1: Training paradigms for GUI grounding and validation of coordinate-token confidence. **Left**: Comparison of training methods: (a) SFT and (b) RLVR require ground truth labels, while (c) our CAL/CANL methods leverage this confidence signal for label-free training. **Right**: Distribution of coordinate-token confidence on ScreenSpot-V2 shows clear separation between correct and incorrect predictions, with correct predictions exhibiting significantly higher confidence values.

To address this challenge, we propose leveraging the model's own confidence signals as a source of supervision. Our key insight is that while model's most confident predictions may not always be correct, its coordinate-level token probabilities provide valuable signals for distinguishing likely correct from incorrect predictions. As illustrated in Fig. 1, we empirically observe that the confidence

distributions of correct and incorrect predictions form distinct, well-separated clusters when focusing on coordinate tokens. This observation motivates our development of coordinate-token confidence, a metric that focuses specifically on the probability values of tokens representing spatial coordinates rather than the entire response sequence.

Building on this foundation, we introduce **C**onfidence-**A**nchored **L**earning (CAL), which generates multiple candidate predictions, selects the most confident one as a pseudo-label based on coordinate-token confidence, and performs reinforcement learning using binary rewards derived from spatial distances to this pseudo-label. As shown in Fig. 1(c), our approach eliminates the need for ground truth labels by using these pseudo-labels to assign rewards directly, enabling truly label-free training. However, pseudo-labels inherently contain errors that could negatively impact learning. This leads to a critical observation: in the sparse coordinate space of GUI grounding, negative samples (predictions far from the pseudo-label) are overwhelmingly likely to be incorrect, while positive samples near potentially misplaced pseudo-labels may be unreliable. This asymmetry motivates **C**onfidence-**A**nchored **N**egative **L**earning (CANL), which modifies the advantage computation to zero out potentially unreliable positive samples while preserving negative learning signals. By focusing solely on reliably incorrect predictions, CANL transforms the challenge of pseudo-label uncertainty into an opportunity for robust learning.

Evaluation across four benchmarks validates our approach. Without any annotations, CANL-7B achieves 92.1% on ScreenSpot-V2, bringing a 4% performance improvement. CANL-7B reaches 33.8% on ScreenSpot-Pro, an 8.9% absolute improvement over the base model, exceeding GUI-R1-7B which requires ground truth rewards. On challenging benchmarks ScreenSpot-Pro and UI-Vision, CANL consistently outperforms CAL by 0.6-1.5%, confirming that negative samples provide more reliable supervision when targets are small and sparse. Analysis reveals coordinate-token confidence outperforms eight alternative pseudo-labeling strategies by 2.1-11.9%, while CANL maintains stable performance across predefined distance thresholds compared to CAL's variance, demonstrating superior robustness for practical deployment. Furthermore, Experiments on AndroidWorld demonstrate that the grounding performance gains delivered by our methods can be effectively applied to GUI navigation tasks.

Our contributions can be summarized as follows:

- We propose CAL, a label-free training approach for GUI grounding that introduces coordinate-token confidence to identify optimal predictions among multiple candidates and uses these as pseudo-labels for reinforcement learning without manual annotations.
- We further develop CANL, which exclusively uses negative samples during reinforcement learning, demonstrating that learning from reliably incorrect predictions can be more effective than using potentially inaccurate positive samples in label-free settings.
- Extensive experiments demonstrate that our label-free approaches achieve competitive performance across GUI Grounding benchmarks.

2 Related Work

2.1 GUI Grounding

GUI grounding bridges natural language instructions with GUI elements, enabling agents to understand and interact with software environments through grounded multimodal reasoning [7, 34, 36]. Given a screenshot and a natural language command, the task requires identifying the corresponding UI element and returning its location as either a bounding box or a point coordinate. Early efforts in enhancing GUI grounding capabilities primarily relied on supervised fine-tuning (SFT) [7, 12, 15, 45, 49] using large-scale annotated datasets. Building on the success of GRPO and DeepSeek-R1 [14, 31], recent work has shifted toward RLVR, where models leverage interaction signals and feedback to further enhance grounding performance [18, 22, 24, 26, 35, 51, 53, 60]. However, both SFT and RLVR paradigms fundamentally depend on labeled data, where manual annotation is prohibitively expensive and automated labeling processes often introduce errors, creating a major bottleneck for scaling GUI agents to diverse applications and domains. GUI-RCPO [9] adopts unsupervised training with predicted bounding boxes, yet erroneous supervision yields marginal gains.

2.2 Negative Learning

Negative learning (NL) reframes supervised learning by training models on what to avoid rather than what to produce. It operates on the principle that model's least confident predictions serve as reliable negative signals, even when its most confident predictions may be incorrect [19]. This approach has proven effective for tasks with noisy labels and in few-shot settings where robust training signals are scarce [43]. Recently, NL has been applied to large language models for mathematical reasoning, using either an increased ratio of negative samples [5, 42] or negative samples exclusively [61]. Despite its success in text-based domains, the application of NL to multimodal models remains unexplored. Our work explore whether learning solely from negative examples can be an effective strategy for complex vision-language tasks such as GUI grounding.

3 Method

We propose a label-free training paradigm for GUI grounding. Our approach consists of two key components: (1) coordinate-token confidence-based pseudo-label generation that selects the most reliable prediction from multiple samples, and (2) distance-based reinforcement learning that assigns rewards without ground truth. We present two variants: **C**onfidence-**A**nchored **L**earning (CAL) that performs policy optimization using both positive and negative rewards, and **C**onfidence-**A**nchored **N**egative **L**earning (CANL) that exclusively learns from negative samples to mitigate pseudo-label errors in the GUI coordinate space.

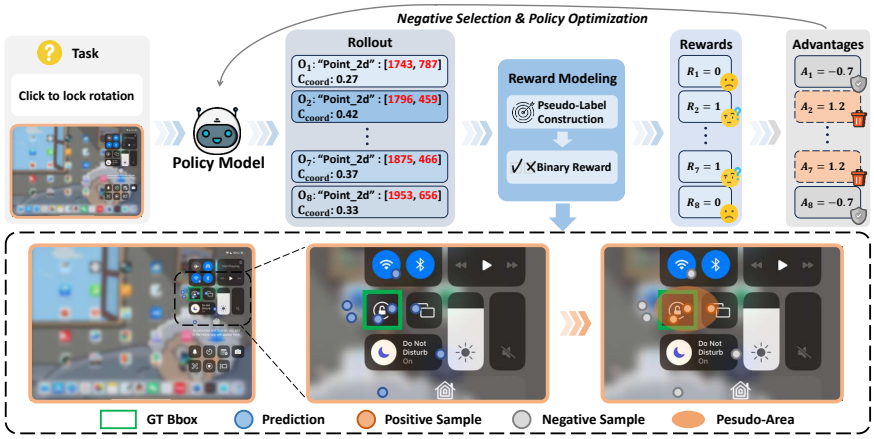

Fig. 2: The CANL pipeline. Given an instruction and screenshot, the model generates multiple predictions with associated coordinate-token confidence scores, selects the highest-confidence prediction as a pseudo-label, constructs a pseudo-area using the distance threshold τ, and assigns binary rewards. During policy optimization, CANL leverages only negative samples (gray) while zeroing advantages for potentially unreliable positive samples (orange), as illustrated in the bottom panel showing the transformation from standard to negative-only reinforcement learning.

3.1 Confidence-Based Pseudo-Label Generation

The core challenge in label-free GUI grounding is identifying reliable training signals from the model's own predictions. Standard confidence estimation averages probabilities across all tokens:

$$c(y \mid x) = \frac{1}{n} \sum_{i=1}^{n} P(y_i \mid y_1, \ldots, y_{i-1}, x) \tag{1}$$

However, this approach fails in GUI tasks as high-probability formatting and descriptive tokens mask the uncertainty inherent in critical coordinate values. Our analysis reveals that coordinate tokens—the numerical values specifying pixel locations—carry the primary uncertainty signal, with 71.5% having probabilities below 0.6 while 78.1% of non-coordinate tokens exceed 0.9 on ScreenSpot-V2. We therefore introduce coordinate-token confidence, which focuses exclusively on these informative tokens:

$$c_{\text{coord}}(y \mid x_{img}, x_{ins}) = \frac{1}{k} \sum_{i \in \text{coord}} P(y_i \mid y_1, \ldots, y_{i-1}, x_{img}, x_{ins}) \tag{2}$$

where k represents the number of coordinate tokens. As shown in Fig. 1 right, this metric effectively distinguishes correct from incorrect predictions. Given an image–instruction pair (x_{img}, x_{ins}), we generate multiple predictions through

parallel sampling and select the one with highest coordinate-token confidence as our pseudo-label:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} c_{\text{coord}}(y \mid x_{img}, x_{ins}), \quad \hat{p} = \text{extract}(\hat{y}) \tag{3}$$

where $\hat{y}$ represents the optimal response selected by coordinate-token confidence, and $\mathcal{Y}$ denotes the collection consisting of generated responses. The function $\text{extract}(y)$ is defined to obtain a point from the response y. $\hat{p}$ indicates the optimal predicted point, i.e. the pseudo-label. This confidence-based selection provides a principled basis for self-supervised learning without requiring ground truth annotations.

3.2 Confidence-Anchored Reinforcement Learning

To enable reinforcement learning without ground truth annotations, we design a distance-based reward mechanism using our pseudo-labels. For the ith sampled prediction (x_i, y_i), given an image with height H and width W, we normalize its coordinates as $(x_i/W, y_i/H)$, and compute its Euclidean distance to the pseudo-label and assign binary rewards:

$$R_i = \begin{cases} 1, & \text{if } \text{dist}(p_i, \hat{p}) \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

This formulation exploits the sparse nature of GUI coordinate space where predictions close to our high-confidence pseudo-label are likely correct, while distant predictions are almost certainly wrong. The distance threshold τ defines the boundary between potentially correct and incorrect predictions. We then apply Group Relative Policy Optimization (GRPO) [31], which estimates advantages through standardization across multiple samples:

$$A_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^N)}{\text{std}(\{R_j\}_{j=1}^N)} \tag{5}$$

where N denotes the sampling number. The policy optimization follows the standard GRPO objective with clipped probability ratios and KL regularization:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,o_i} \frac{1}{G} \sum_{i=1}^{G} [\min\left(r_i(\theta)A_i, \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)A_i\right) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta | \pi_{\text{ref}}]] \tag{6}$$

where the probability ratio $r_i(\theta)$ prevents excessive policy updates, β controls the strength of regularization, and G denotes the group size. By treating the model's most confident prediction as a learning target and using spatial proximity to define rewards, CAL transforms GUI grounding into a self-supervised reinforcement learning problem requiring no external annotations.

3.3 Confidence-Anchored Negative Reinforcement Learning

While CAL provides a path to label-free training, pseudo-labels inevitably contain errors that can mislead learning, particularly when incorrect predictions are mistakenly rewarded as positive examples. However, the sparse nature of GUI coordinate space offers an opportunity: negative samples are overwhelmingly reliable since predictions far from any reasonable target are almost certainly incorrect. CANL exploits this asymmetry by modifying the advantage computation to use only negative samples:

$$A_i = \begin{cases} 0, & \text{if } R_i = 1 \\ \frac{-\text{mean}(\{R_j\}_{j=1}^N)}{\text{std}(\{R_j\}_{j=1}^N)}, & \text{if } R_i = 0 \end{cases} \tag{7}$$

This approach zeros out advantages for potentially unreliable positive samples while preserving the learning signal from negative samples. The model thus learns exclusively from what to avoid rather than what to produce, which proves particularly effective in high-resolution GUI tasks where the vast coordinate space makes distant predictions almost certainly incorrect. As illustrated in Fig. 2, CANL maintains the same pseudo-label generation and reward assignment pipeline as CAL but selectively updates the model using only the most reliable learning signals. This strategy effectively leverages label uncertainty as a catalyst for robust negative learning, yielding superior performance on complex datasets where pseudo-label quality is often vulnerable to noise.

4 Experiments

4.1 Experiments Setup

Datasets and Benchmarks. We evaluate on four benchmarks: ScreenSpot [7] and ScreenSpot-V2 [45], which have low-resolution interfaces and larger targets, and more challenging benchmark ScreenSpot-Pro [20] and UI-Vision [27]. For UI-Vision, we only use the Element Grounding subset for training and evaluation, without Layout Grounding and Action Prediction subset. Predictions are correct if they fall within ground truth bounding boxes.

Implementation Details. We use Qwen-2.5-VL [4] (3B/7B) as the base model within the VLM-R1 framework [32]. Following [62], we adopt a test-time training setting where the model is optimized and evaluated on each benchmark independently, obviating the need for a separate training set. All trainings are conducted for 1 epoch, with learning rate 1e-6. To enhance diversity during sampling, we set temperature $T = 1.0$, $top\_k = 50$, $top\_p = 1.0$. KL penalty β is set to 0.04. Distance threshold τ is set to 0.05. We apply Flash Attention2 [8] for training. Following [62], to reduce computational costs while improve the accuracy of pseudo-labels, we apply downsampling strategy during training. Specifically, we sample 16 responses to construct pseudo labels. For CAL, we randomly downsample 8 responses to compute the advantages. For CANL, we compute the

Table 1: Performance comparison on ScreenSpot-V1 and V2. "-" indicates missing values due to unavailable results, unreleased checkpoints, and code. For label-free methods, the optimal and the suboptimal results are **bolded** and underlined, respectively.

Model	GUI Labels	v1 Mobile		v1 Desktop		v1 Web		v1 Avg.	v2 Avg.
		Text	Icon	Text	Icon	Text	Icon		
Proprietary Models									
GPT-4o [28]	-	30.5	23.2	20.6	19.4	11.1	7.8	18.8	20.1
Claude Computer Use [1]	-	-	-	-	-	-	-	83.0	-
General Models									
Qwen-2.5-VL-3B [4]	0	93.8	68.1	91.2	55.0	81.7	64.6	77.6	82.1
Qwen-2.5-VL-7B [4]	0	91.9	80.8	88.1	75.7	90.0	77.7	84.9	88.1
GUI-specific Models (Label-required)									
CogAgent-18B [15]	222M	67.0	24.0	74.2	20.0	70.4	28.6	47.4	-
SeeClick-9.6B [7]	1M	78.0	52.0	72.2	30.0	55.7	32.5	53.4	55.1
UGround-7B [12]	10M	82.8	60.3	82.5	63.6	80.4	70.4	73.3	76.3
OS-Atlas-7B [45]	13M	93.0	72.9	91.8	62.9	90.9	74.3	82.5	-
ShowUI-2B [21]	256K	92.3	75.5	76.3	61.1	81.7	63.6	75.1	77.3
Aguvis-72B [49]	1M	94.5	85.2	95.4	77.9	91.3	85.9	89.2	-
UI-TARS-7B [29]	18.4M	94.5	85.2	95.9	85.7	90.0	83.5	89.5	91.6
UI-TARS-72B [29]	18.4M	94.9	82.5	89.7	88.6	88.7	85.0	88.4	90.3
GUI-Actor-7B [44]	9.6M	94.9	82.1	91.8	80.0	91.3	85.4	88.3	92.1
Jedi-7B [46]	4M	-	-	-	-	-	-	-	91.7
UI-R1-3B [24]	136	95.6	84.7	90.2	59.3	85.2	73.3	83.3	85.4
GUI-R1-7B [26]	3K	-	-	91.8	73.6	91.3	75.7	-	-
InfiGUI-R1-3B [22]	32K	97.1	81.2	94.3	77.1	91.7	77.6	87.5	-
SE-GUI-7B [53]	3K	-	-	-	-	-	-	88.2	90.3
GuirlVG-7B [18]	5.2K	96.0	84.7	92.8	80.0	92.6	85.9	88.7	91.9
GUI-specific Models (Label-free)									
GUI-RCPO-7B [9]	0	-	-	-	-	-	-	86.6	88.9
Ours									
CAL-3B	0	96.7	78.6	95.4	67.9	87.8	72.8	84.6	88.9
CANL-3B	0	96.0	79.0	**96.4**	66.4	87.0	73.8	84.5	88.5
CAL-7B	0	**97.1**	**87.3**	86.1	80.7	**91.7**	83.0	88.6	**92.1**
CANL-7B	0	96.7	**87.3**	88.6	**82.1**	91.3	**84.0**	**89.2**	**92.1**

advantages using 16 responses and then select the 8 negative responses farthest from the pseudo-label for policy optimization.

Baselines. We compare against three categories of methods: (1) Label-required models, including SFT paradigms (e.g., SeeClick [7], ShowUI [21], and UI-TARS [29]) and RL approaches (e.g., UI-R1 [24], GUI-R1 [26]); (2) Label-free model: GUI-RCPO [9], which leverages region consistency for training guidance; (3) Proprietary models such as GPT-4o [28] and Claude Computer Use [1]. Both our method and GUI-RCPO are trained on test sets, enabling a fair performance comparison. By contrast, comparisons with label-required models mainly highlight our data efficiency.

4.2 Main Results

Our label-free methods achieve competitive performance without any annotations. Table 1 demonstrates substantial improvements over vanilla base models on ScreenSpot and ScreenSpot-V2, with absolute gains ranging from 4.0% to 7%. On ScreenSpot-V2, CANL-7B achieves 92.1%, improving 4.0% over the base Qwen-2.5-VL-7B, outperforming GUI-RCPO-7B (88.9%), proving the

Table 2: Performance comparison on ScreenSpot-Pro. For label-free methods, the optimal and the suboptimal results are **bolded** and underlined, respectively.

Model	GUI Labels	CAD		Dev		Creative		Scientific		Office		OS		Avg.
		Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Proprietary Models														
GPT-4o [28]	-	2.0	0.0	1.3	0.0	1.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	0.8
Claude Computer Use [1]	-	14.5	3.7	22.0	3.9	25.9	3.4	33.9	15.8	30.1	16.3	11.0	4.5	17.1
General Models														
Qwen-2.5-VL-3B [4]	0	9.1	7.3	22.1	1.4	26.8	2.1	38.2	7.3	33.9	15.1	10.3	1.1	16.1
Qwen-2.5-VL-7B [4]	0	13.7	7.8	44.2	6.9	28.8	10.5	48.6	5.5	46.9	15.1	31.8	12.4	24.9
GUI-specific Models (Label-required)														
SeeClick-9.6B [7]	1M	2.5	0.0	0.6	0.0	1.0	0.0	3.5	0.0	1.1	0.0	2.8	0.0	1.1
CogAgent-18B [15]	222M	7.1	3.1	14.9	0.7	9.6	0.0	22.2	1.8	13.0	0.0	5.6	0.0	7.7
Aria-UI [52]	17.6M	7.6	1.6	16.2	0.0	23.7	2.1	27.1	6.4	20.3	1.9	4.7	0.0	11.3
OS-Atlas-7B [45]	13M	12.2	4.7	33.1	1.4	28.8	2.8	37.5	7.3	33.9	5.7	27.1	4.5	18.9
ShowUI-2B [21]	256K	2.5	0.0	16.9	1.4	9.1	0.0	13.2	7.3	15.3	7.5	10.3	2.2	7.7
UGround-7B [12]	10M	14.2	1.6	26.6	2.1	27.3	2.8	31.9	2.7	31.6	11.3	17.8	0.0	16.5
ZonUI-3B [16]	24K	31.9	15.6	24.6	6.2	40.9	7.6	54.8	18.1	57.0	26.4	19.6	7.8	28.7
UI-R1-3B [24]	136	11.2	6.3	22.7	4.1	27.3	3.5	42.4	11.8	32.2	11.3	13.1	4.5	17.8
GUI-R1-7B [26]	3K	23.9	6.3	49.4	4.8	38.9	8.4	55.6	11.8	58.7	26.4	42.1	16.9	31.0
UI-Ins-32B [6]	316K	51.8	29.7	83.1	26.9	69.7	18.9	83.3	34.5	88.7	50.9	70.1	34.8	57.0
GUI-specific Models (Label-free)														
GUI-RCPO-7B [9]	0	-	-	-	-	-	-	-	-	-	-	-	-	25.9
Ours														
CAL-3B	0	35.0	9.4	44.2	3.4	**46.5**	4.9	60.4	18.1	53.1	20.8	38.3	7.9	32.1
CANL-3B	0	**39.1**	7.8	42.9	4.8	46.0	3.5	58.3	**20.9**	55.9	22.6	38.3	7.9	32.7
CAL-7B	0	25.4	**11.0**	51.3	6.9	38.9	**9.8**	60.4	15.5	**64.7**	19.2	37.4	**19.2**	32.7
CANL-7B	0	29.4	10.9	**53.2**	**9.0**	37.4	8.4	**63.2**	14.5	62.1	**26.4**	**40.2**	15.7	**33.8**

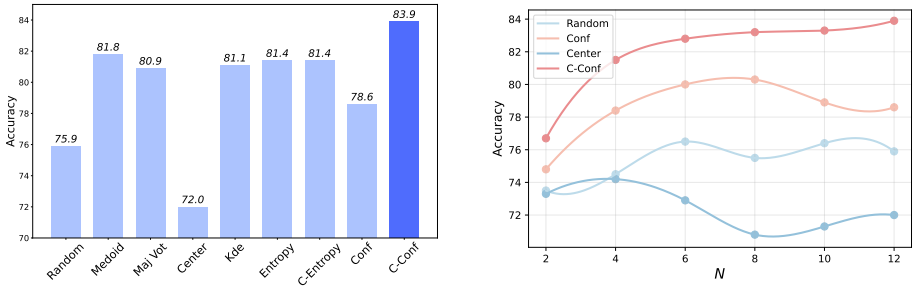

Fig. 3: Performance of different pseudo-label construction methods. **Left**: Comparison of pseudo-label construction methods with sampling number N=12. **Right**: Pseudo-label quality scales with sampling budget across four strategies.

reward signal of our methods is more reliable. Meanwhile, CAL-7B matches GUI-Actor-7B (92.1%) which uses 9.6M labels. At the 3B scale, CANL improves the base model from 77.6% to 84.5% (+6.9%), approaching InfiGUI-R1-3B (87.5%) which requires 32K labels, validating that coordinate-token confidence provides sufficient supervision.

Negative learning dominates on challenging high-resolution benchmarks. As shown in Tab. 2 and Tab. 3, the performance gap between CAL

Table 3: Performance comparison on UI-Vision. For label-free methods, the optimal and the suboptimal results are **bolded** and underlined, respectively.

Model	GUI Labels	Grouped by Category						Grouped by Setting			Overall
		Edu.	Browser	Dev.	Prod.	Creative	Entert.	Basic	Func.	Spatial	
Proprietary Models											
GPT-4o [28]	-	1.5	0.0	2.2	1.1	0.8	4.2	1.6	1.5	1.0	1.4
Claude Computer Use [1]	-	6.1	9.8	8.0	9.4	7.7	8.3	9.5	7.7	7.6	8.3
General Models											
Qwen-2.5-VL-3B [4]	0	7.6	22.4	15.4	12.8	6.6	33.9	18.6	13.8	4.3	12.0
Qwen-2.5-VL-7B [4]	0	11.1	37.1	18.1	15.4	9.6	29.7	20.0	18.6	7.1	15.0
GUI-specific Models (Label-required)											
SeeClick-9.6B [7]	1M	4.2	13.3	7.3	4.3	4.0	11.0	9.4	4.7	2.1	5.4
ShowUI-2B [21]	256K	3.7	13.3	7.5	6.5	2.5	15.6	8.1	7.7	2.1	5.9
CogAgent-9B [15]	222M	8.7	11.2	8.6	10.3	5.6	15.6	12.0	12.2	2.6	8.9
OSAtlas-7B [45]	13M	8.7	16.8	10.3	9.2	5.6	16.2	12.2	11.2	3.7	9.0
AriaUI [52]	17.6M	9.0	18.9	11.2	10.4	6.5	19.3	12.2	14.0	4.0	10.1
UGround-v1-7B [12]	-	10.4	28.7	17.5	12.2	8.6	18.2	15.4	17.1	6.3	12.9
Aguvis-7B [49]	1M	13.1	30.8	17.1	12.1	9.6	24.0	17.8	18.3	5.1	13.7
UI-TARS-7B [29]	18.4M	14.2	35.0	19.7	18.3	11.1	38.5	20.1	24.3	8.4	17.6
Ours											
CAL-3B	0	15.1	32.9	20.6	18.1	10.4	37.0	24.9	22.1	5.7	17.2
CANL-3B	0	16.0	33.6	21.6	19.4	12.0	39.1	26.4	23.8	6.5	18.5
CAL-7B	0	**17.6**	**42.7**	22.0	18.9	10.9	41.1	25.6	22.9	8.4	18.6
CANL-7B	0	17.3	38.5	**24.1**	**20.9**	**12.4**	**44.8**	**27.5**	**25.1**	**8.8**	**20.1**

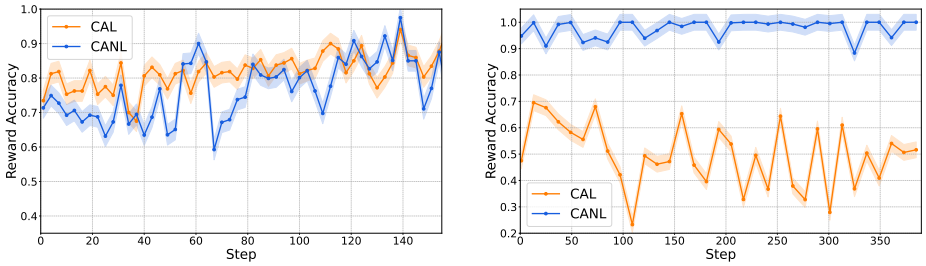

Fig. 4: Left: Reward accuracy on ScreenSpot-V2. **Right**: Reward accuracy on ScreenSpot-Pro.

and CANL reveals a critical pattern: as difficulty increases, exclusive negative learning becomes increasingly advantageous. On ScreenSpot-Pro, CANL-7B achieves 33.8%, improving 1.1% over CAL-7B, 8.9% over the base model, and GUI-RCPO-7B (25.9%) by a notable margin. This advantage extends to UI-Vision where CANL-7B reaches 20.1%, surpassing CAL by 1.5%. The systematic superiority of CANL confirms that on high-resolution and challenging tasks, negative samples provide more reliable learning signals than potentially incorrect positive samples derived from pseudo-labels.

4.3 In-depth Analysis

Coordinate-token confidence outperforms alternative pseudo-labeling strategies. Figure 3 evaluates different test-time scaling methods to generate

pseudo-label on ScreenSpot-V2. Detailed descriptions of these methods are provided in Appendix A.5. Our proposed coordinate-token confidence (C-Conf) achieves 83.9% accuracy with $N = 12$ samples, substantially exceeding alternatives. The 5.3% gain over standard confidence validates our core insight that coordinate tokens carry the primary uncertainty signal in GUI grounding, as formatting and descriptive tokens exhibit consistently high probabilities regardless of prediction quality. Moreover, this superiority persists across varying sampling budgets: C-Conf maintains its advantage from minimal sampling ($N = 2$: 76.7% vs 74.8% for standard confidence) to larger ensembles ($N = 12$: 83.9% vs 78.6%). Notably, while spatial methods like Center collapse at higher sampling rates due to averaging effects, C-Conf shows monotonic improvement, confirming that confidence-based selection scales effectively with computational budget.

Reward reliability validates the asymmetric quality of positive and negative samples. Figure 4 quantifies the actual correctness of our binary reward assignments by comparing against ground truth labels. On ScreenSpot-V2 with larger bounding boxes, CANL's reward accuracy falls slightly below CAL's because some correct predictions lying outside threshold τ are misclassified as negative. However, CANL maintains near-perfect negative sample reliability (>0.95) while CAL's mixed positive-negative accuracy hovers around 0.5. This empirical validation confirms our core hypothesis that high-resolution GUI tasks naturally provide accurate negative signals even without labels, as the vast coordinate space makes random distant predictions almost certainly incorrect.

Distance threshold sensitivity reveals fundamental differences between positive and negative learning. Table 4 examines how the threshold τ affects both methods across datasets. On ScreenSpot-V2, CAL exhibits strong sensitivity with performance degrading from 88.9% at $\tau = 0.05$ to 86.9% at $\tau = 0.1$, reflecting its reliance on accurate positive sample identification. Conversely, CANL maintains remarkable stability at 88.5%±1.0% across all thresholds. This pattern intensifies on ScreenSpot-Pro where CAL's performance drops precipitously from 33.2% to 30.1% as τ increases, while CANL remain stable. This invariance stems from a key property of sparse coordinate spaces: predictions far from any reasonable target remain unambiguously incorrect as distance threshold increases. While positive samples require careful boundary definition to avoid including incorrect predictions, negative samples beyond any plausible threshold provide consistently reliable learning signals, making CANL robust to hyperparameter.

Table 4: Distance threshold τ sensitivity analysis.

τ	SS-V2	SS-Pro
CAL		
0.01	87.7	33.3
0.03	88.4	33.2
0.05	**88.9**	33.2
0.07	**88.9**	30.1
0.1	86.9	30.7
CANL		
0.01	87.5	30.2
0.03	87.8	30.8
0.05	<u>88.5</u>	**33.7**
0.07	88.4	32.7
0.1	<u>88.5</u>	<u>33.5</u>

Training dynamics reveal distinct convergence patterns across difficulty levels. Figure 5 tracks the evolution of positive ratio, defined as the frac-

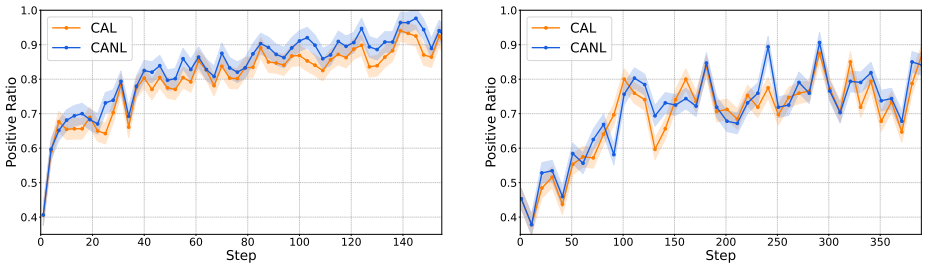

Fig. 5: Left: Positive Ratio on ScreenSpot-V2. **Right**: Positive Ratio on ScreenSpot-Pro.

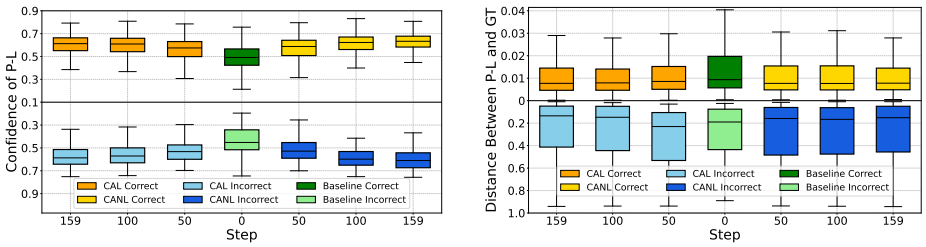

Fig. 6: Left: Confidence of pseudo-label on ScreenSpot-V2. **Right**: Distance between the pseudo-label and the center of ground truth bounding box on ScreenSpot-V2.

tion of samples within distance threshold τ from the pseudo-label. On ScreenSpot-V2, both methods exhibit monotonic increases, indicating progressive prediction concentration around confident regions. ScreenSpot-Pro presents markedly different dynamics: slower convergence with higher variance, plateauing around 0.8 rather than continuing upward. This divergence reflects the inherently greater challenge of high-resolution grounding where precise localization becomes critical. Notably, CANL consistently maintains higher positive ratios than CAL across both settings, suggesting that learning exclusively from negative samples paradoxically helps models develop more concentrated predictions, possibly by more effectively pruning incorrect hypotheses from the prediction space.

Pseudo-label characteristics reveal both successes and inherent limitations. Figure 6 provides deeper insights into pseudo-label behavior. The left panel shows coordinate-token confidence increases for both correct and incorrect pseudo-labels during training, with correct predictions maintaining a consistent advantage. This convergence suggests models become increasingly confident regardless of accuracy, potentially limiting further improvements without external supervision. The right panel reveals a striking bimodal distribution: correct pseudo-labels cluster within 0.02 normalized distance of ground truth centers, while incorrect ones remain at 0.40 distance throughout training. This binary pattern validates our distance-based reward design and explains why negative

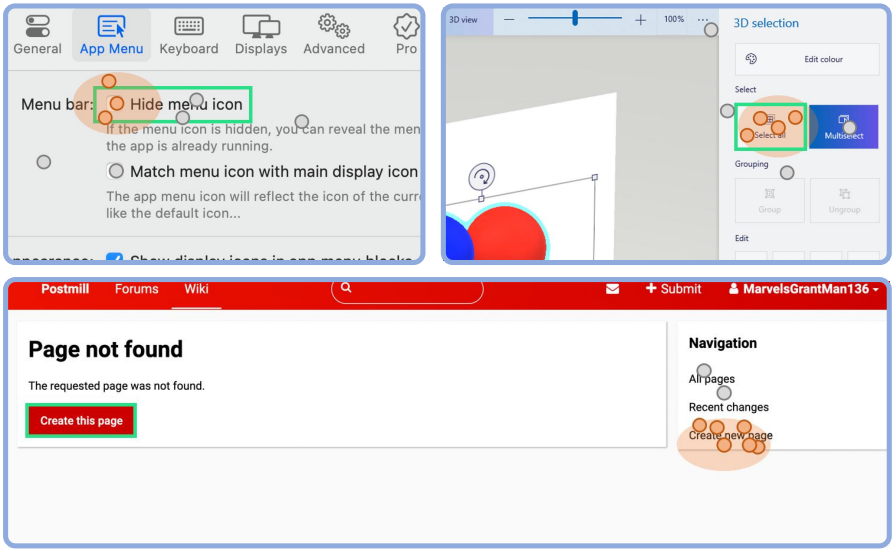

Fig. 7: Cropped images of different cases. **Top-Left**: Geometric mismatch. **Top-Right**: Good alignment. **Bottom**: Misplaced pseudo-label.

samples are overwhelmingly reliable in sparse coordinate spaces. The persistence of distant incorrect pseudo-labels indicates certain challenging cases remain beyond the model's capability without ground truth, representing an inherent limitation of label-free learning.

Reward estimation cases demonstrate the advantages of negative learning. To assess the reliability of our distance-based reward assignment, we analyze three common scenarios during pseudo-label generation, shown in Fig. 7: **Geometric mismatch**: The pseudo-label is within the target bounding box, but the ground truth's elongated shape does not match the region generated by the pseudo-labels. **Good alignment**: The pseudo-label and reward region align well with the target, resulting in accurate reward estimation. **Misplaced pseudo-label**: The pseudo-label is far from the target, causing all samples in the reward region to be incorrectly rewarded, though negative samples remain correctly classified. These cases highlight the varying reliability of positive samples under different pseudo-label conditions and explain why CANL's focus on negative samples yields more robust learning signals.

Cross-dataset experiments demonstrate the generalization capability. We train Qwen-2.5-VL-3B for one epoch on a source dataset and evaluate its transferability, as shown in Fig. 8; ScreenSpot-V1 is excluded due to documented annotation errors [45]. Training on UI-Vision yields gains of 5.3% on ScreenSpot-V2 (87.4% vs. 82.1%) and 15.6% on ScreenSpot-Pro (31.7% vs. 16.1%), while training on ScreenSpot-V2 improves UI-Vision by 4.6%. These results confirm that our method avoids overfitting, achieving robust performance on unseen dis-

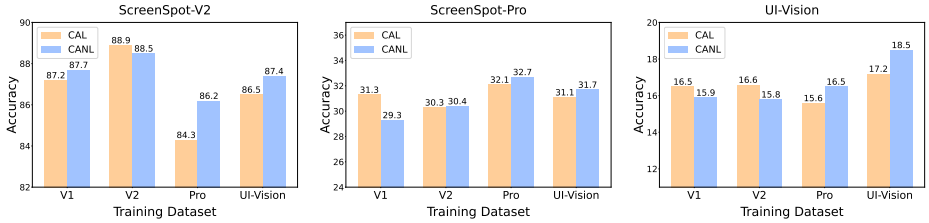

Fig. 8: Cross-dataset generalization performance. Accuracy of our methods across different training sets when evaluated on ScreenSpot-V2, ScreenSpot-Pro, and UI-Vision.

tributions. Notably, on more challenging benchmarks, CANL consistently outpaces CAL, highlighting its superior robustness in complex scenarios.

Training on public datasets further validates generalization. we extend our training to GroundCUA [10], a publicly available benchmark. Specifically, we randomly sample 5k instances for a single epoch of training, with results summarized in Tab. 5. Our method exhibits robust generalization capability; even when trained on external public data, it yields substantial performance gains across evaluation benchmarks. This underscores that our label-free paradigm effectively captures the underlying spatial logic of GUI grounding rather than merely memorizing dataset-specific patterns.

Table 5: Evaluation of our methods trained on publicly available datasets.

Method	ScreenSpot	ScreenSpot-V2	ScreenSpot-Pro	UI-Vision
Qwen-2.5-VL-3B	77.6	82.1	16.1	12.0
w/CAL	**84.6**	**87.3**	32.6	**16.7**
w/CANL	83.5	86.7	**33.4**	**16.7**

Online end-to-end evaluation demonstrates the effectiveness of our method in real-world scenarios. We evaluate our approach on AndroidWorld [30], a realistic platform comprising 116 tasks. To isolate the impact on GUI grounding, we adopt the SeeAct-V framework [12], which decouples planning from precise coordinate prediction. Specifically, we evaluate the model previously trained on ScreenSpot-V2 [45] using our methods. As shown in Tab. 6, our approaches yield a 4.3–5.2% performance gain. CAL slightly outperforms CANL, since the simple training dataset produces more precise pseudo-

Table 6: Success rate on AndroidWorld.

Planner	Grounding	SR
GPT-4o	Qwen-2.5-VL-3B	25.0
	w/CAL	**30.2**
	w/CANL	29.3

labels. These results indicate that the enhanced grounding precision from our methods directly mitigates execution failures in multi-step sequences, demonstrating the practical efficacy of our methods for real-world GUI agents.

5 Conclusion and Future Improvement

This work introduces a label-free paradigm that successfully decouples GUI grounding from expensive manual labeling. By identifying coordinate-token confidence as a reliable proxy for accuracy, we proposed CAL and CANL to enable autonomous optimization. Our results—particularly the success of CANL—show that in sparse GUI environments, negative reinforcement can be more effective than error-prone positive pseudo-labels. However, an inherent limitation remains: while negative samples are highly reliable in challenging scenarios, the absence of accurately positive reward signal—compared to supervised training—may constrain the further scaling of model capabilities. Future research could focus on integrating more accurate positive feedback to achieve a more synergistic balance between avoiding errors and reinforcing optimal grounding behaviors.

Acknowledgement

This work was supported by New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123100), National Natural Science Foundation of China (No. 62506332), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (NO. 2026C02A1223), and CCF-Ant-Research Fund.

References

1. Anthropic: Claude computer use. Available at: https://www.anthropic.com/news/developing-computer-use (2024), accessed 27 June 2026
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 (2023)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
5. Chen, H., Zheng, K., Zhang, Q., Cui, G., Cui, Y., Ye, H., Lin, T.Y., Liu, M.Y., Zhu, J., Wang, H.: Bridging supervised learning and reinforcement learning in math reasoning. arXiv preprint arXiv:2505.18116 (2025)
6. Chen, L., Zhou, H., Cai, C., Zhang, J., Tong, P., Kong, Q., Zhang, X., Liu, C., Liu, Y., Wang, W., et al.: Ui-ins: Enhancing gui grounding with multi-perspective instruction-as-reasoning. arXiv preprint arXiv:2510.20286 (2025)

7. Cheng, K., Sun, Q., Chu, Y., Xu, F., Li, Y., Zhang, J., Wu, Z.: Seeclick: Harnessing gui grounding for advanced visual gui agents. arXiv preprint arXiv:2401.10935 (2024)
8. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)
9. Du, Y., Yan, Y., Tang, F., Lu, Z., Zong, C., Lu, W., Jiang, S., Shen, Y.: Test-time reinforcement learning for gui grounding via region consistency (2025)
10. Feizi, A., Nayak, S., Jian, X., Lin, K.Q., Li, K., Awal, R., Lù, X.H., Obando-Ceron, J., Rodriguez, J.A., Chapados, N., et al.: Grounding computer use agents on human demonstrations. arXiv preprint arXiv:2511.07332 (2025)
11. Gao, L., Zhang, L., Gao, P., Liu, W., Luan, J., Xu, M.: Gui-shift: Enhancing vlm-based gui agents through self-supervised reinforcement learning. arXiv preprint arXiv:2505.12493 (2025)
12. Gou, B., Wang, R., Zheng, B., Xie, Y., Chang, C., Shu, Y., Sun, H., Su, Y.: Navigating the digital world as humans do: Universal visual grounding for gui agents. arXiv preprint arXiv:2410.05243 (2024)
13. Gu, Z., Zeng, Z., Xu, Z., Zhou, X., Shen, S., Liu, Y., Zhou, B., Meng, C., Xia, T., Chen, W., et al.: Ui-venus technical report: Building high-performance ui agents with rft. arXiv preprint arXiv:2508.10833 (2025)
14. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
15. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., et al.: Cogagent: A visual language model for gui agents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14281–14290 (2024)
16. Hsieh, Z., Wei, T.J., Yang, S.: Zonui-3b: A lightweight vision-language model for cross-resolution gui grounding. arXiv e-prints pp. arXiv–2506 (2025)
17. Hu, X., Xiong, T., Yi, B., Wei, Z., Xiao, R., Chen, Y., Ye, J., Tao, M., Zhou, X., Zhao, Z., et al.: Os agents: A survey on mllm-based agents for computer, phone and browser use. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 7436–7465 (2025)
18. Kang, W., Lei, B., Liu, G., Ding, C., Yan, Y.: Guirlvg: Incentivize gui visual grounding via empirical exploration on reinforcement learning. arXiv preprint arXiv:2508.04389 (2025)
19. Kim, Y., Yim, J., Yun, J., Kim, J.: Nlnl: Negative learning for noisy labels. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 101–110 (2019)
20. Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., Ma, J., Huang, Z., Chua, T.S.: Screenspot-pro: Gui grounding for professional high-resolution computer use. arXiv preprint arXiv:2504.07981 (2025)
21. Lin, K.Q., Li, L., Gao, D., Yang, Z., Wu, S., Bai, Z., Lei, S.W., Wang, L., Shou, M.Z.: Showui: One vision-language-action model for gui visual agent. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 19498–19508 (2025)
22. Liu, Y., Li, P., Xie, C., Hu, X., Han, X., Zhang, S., Yang, H., Wu, F.: Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. arXiv preprint arXiv:2504.14239 (2025)
23. Liu, Y., Liu, Z., Zhu, S., Li, P., Xie, C., Wang, J., Hu, X., Han, X., Yuan, J., Wang, X., et al.: Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. arXiv preprint arXiv:2508.05731 (2025)

24. Lu, Z., Chai, Y., Guo, Y., Yin, X., Liu, L., Wang, H., Xiao, H., Ren, S., Xiong, G., Li, H.: Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. arXiv preprint arXiv:2503.21620 (2025)
25. Lu, Z., Ye, J., Tang, F., Shen, Y., Xu, H., Zheng, Z., Lu, W., Yan, M., Huang, F., Xiao, J., et al.: Ui-s1: Advancing gui automation via semi-online reinforcement learning. arXiv preprint arXiv:2509.11543 (2025)
26. Luo, R., Wang, L., He, W., Xia, X.: Gui-r1: A generalist r1-style vision-language action model for gui agents. arXiv preprint arXiv:2504.10458 (2025)
27. Nayak, S., Jian, X., Lin, K.Q., Rodriguez, J.A., Kalsi, M., Awal, R., Chapados, N., Özsu, M.T., Agrawal, A., Vazquez, D., et al.: Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. arXiv preprint arXiv:2503.15661 (2025)
28. OpenAI: Introducing gpt-4o. Available at: https://openai.com/index/hello-gpt-4o (2024), accessed 27 June 2026
29. Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al.: Ui-tars: Pioneering automated gui interaction with native agents. arXiv preprint arXiv:2501.12326 (2025)
30. Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W., Li, W., Campbell-Ajala, F., et al.: Androidworld: A dynamic benchmarking environment for autonomous agents. arXiv preprint arXiv:2405.14573 (2024)
31. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
32. Shen, H., Liu, P., Li, J., Fang, C., Ma, Y., Liao, J., Shen, Q., Zhang, Z., Zhao, K., Zhang, Q., et al.: Vlm-r1: A stable and generalizable r1-style large vision-language model. arXiv preprint arXiv:2504.07615 (2025)
33. 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., Wu, Z.: Os-genesis: Automating gui agent trajectory construction via reverse task synthesis. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2025)
34. Tang, F., Chen, B., Lu, Z., Chen, T., Nong, S., Jiang, T., Xu, W., Lu, W., Xiao, J., Zhuang, Y., et al.: Ui-zoomer: Uncertainty-driven adaptive zoom-in for gui grounding. arXiv preprint arXiv:2604.14113 (2026)
35. Tang, F., Gu, Z., Lu, Z., Liu, X., Shen, S., Meng, C., Wang, W., Zhang, W., Shen, Y., Lu, W., et al.: Gui-g ^{2}: Gaussian reward modeling for gui grounding. arXiv preprint arXiv:2507.15846 (2025)
36. Tang, F., Gu, Z., Lu, Z., Zhang, S., Zeng, Z., Shen, S., Meng, C., Yan, Y., Zhang, W., Shen, Y., et al.: Gui-sage: Enhancing gui automation with self-explanatory learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13007–13016 (2026)
37. Tang, F., Lu, Z., Zhang, B., Lu, W., Xiao, J., Zhuang, Y., Shen, Y.: Clawgui: A unified framework for training, evaluating, and deploying gui agents. arXiv preprint arXiv:2604.11784 (2026)
38. Tang, F., Shen, Y., Zhang, H., Chen, S., Hou, G., Zhang, W., Zhang, W., Song, K., Lu, W., Zhuang, Y.: Think twice, click once: Enhancing gui grounding via fast and slow systems. arXiv preprint arXiv:2503.06470 (2025)
39. Tang, F., Xu, H., Zhang, H., Chen, S., Wu, X., Shen, Y., Zhang, W., Hou, G., Tan, Z., Yan, Y., et al.: A survey on (m) llm-based gui agents. arXiv preprint arXiv:2504.13865 (2025)

40. Tang, J., Xia, Y., Wu, Y.F., Hu, Y., Chen, Y., Chen, Q.G., Xu, X., Wu, X., Lu, H., Ma, Y., et al.: Lpo: Towards accurate gui agent interaction via location preference optimization. arXiv preprint arXiv:2506.09373 (2025)
41. Wang, J., Xu, H., Ye, J., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J.: Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. arXiv preprint arXiv:2401.16158 (2024)
42. Wang, R., Li, H., Han, X., Zhang, Y., Baldwin, T.: Learning from failure: Integrating negative examples when fine-tuning large language models as agents. arXiv preprint arXiv:2402.11651 (2024)
43. Wei, X.S., Xu, H.Y., Zhang, F., Peng, Y., Zhou, W.: An embarrassingly simple approach to semi-supervised few-shot learning. Advances in Neural Information Processing Systems **35**, 14489–14500 (2022)
44. Wu, Q., Cheng, K., Yang, R., Zhang, C., Yang, J., Jiang, H., Mu, J., Peng, B., Qiao, B., Tan, R., et al.: Gui-actor: Coordinate-free visual grounding for gui agents. arXiv preprint arXiv:2506.03143 (2025)
45. Wu, Z., Wu, Z., Xu, F., Wang, Y., Sun, Q., Jia, C., Cheng, K., Ding, Z., Chen, L., Liang, P.P., et al.: Os-atlas: A foundation action model for generalist gui agents. arXiv preprint arXiv:2410.23218 (2024)
46. Xie, T., Deng, J., Li, X., Yang, J., Wu, H., Chen, J., Hu, W., Wang, X., Xu, Y., Wang, Z., et al.: Scaling computer-use grounding via user interface decomposition and synthesis. arXiv preprint arXiv:2505.13227 (2025)
47. Xu, H., Zhang, X., Liu, H., Wang, J., Zhu, Z., Zhou, S., Hu, X., Gao, F., Cao, J., Wang, Z., et al.: Mobile-agent-v3. 5: Multi-platform fundamental gui agents. arXiv preprint arXiv:2602.16855 (2026)
48. Xu, Y., Liu, X., Liu, X., Fu, J., Zhang, H., Jing, B., Zhang, S., Wang, Y., Zhao, W., Dong, Y.: Mobilerl: Online agentic reinforcement learning for mobile gui agents. arXiv preprint arXiv:2509.18119 (2025)
49. Xu, Y., Wang, Z., Wang, J., Lu, D., Xie, T., Saha, A., Sahoo, D., Yu, T., Xiong, C.: Aguvis: Unified pure vision agents for autonomous gui interaction. arXiv preprint arXiv:2412.04454 (2024)
50. Yan, H., Wang, J., Huang, X., Shen, Y., Meng, Z., Fan, Z., Tan, K., Gao, J., Shi, L., Yang, M., et al.: Step-gui technical report. arXiv preprint arXiv:2512.15431 (2025)
51. Yang, Y., Li, D., Dai, Y., Yang, Y., Luo, Z., Zhao, Z., Hu, Z., Huang, J., Saha, A., Chen, Z., et al.: Gta1: Gui test-time scaling agent. arXiv preprint arXiv:2507.05791 (2025)
52. Yang, Y., Wang, Y., Li, D., Luo, Z., Chen, B., Huang, C., Li, J.: Aria-ui: Visual grounding for gui instructions. arXiv preprint arXiv:2412.16256 (2024)
53. Yuan, X., Zhang, J., Li, K., Cai, Z., Yao, L., Chen, J., Wang, E., Hou, Q., Chen, J., Jiang, P.T., et al.: Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. arXiv preprint arXiv:2505.12370 (2025)
54. Zhang, M., Xu, Z., Zhu, J., Dai, Q., Qiu, K., Yang, Y., Luo, C., Chen, T., Wagle, J., Franklin, T., et al.: Phi-ground tech report: Advancing perception in gui grounding. arXiv preprint arXiv:2507.23779 (2025)
55. Zhang, Y., Ni, B., Chen, X.S., Zhang, H.R., Rao, Y., Peng, H., Lu, Q., Hu, H., Guo, M.H., Hu, S.M.: Bee: A high-quality corpus and full-stack suite to unlock advanced fully open mllms. arXiv preprint arXiv:2510.13795 (2025)
56. Zhao, Z., Liu, Y., Liu, Y., Wang, H., Tian, L., Zhou, X., You, Y., Yu, Z., Yu, Y., Zhou, J.: Points-gui-g: Gui-grounding journey. arXiv preprint arXiv:2602.06391 (2026)

57. Zhou, H., Zhang, X., Tong, P., Zhang, J., Chen, L., Kong, Q., Cai, C., Liu, C., Wang, Y., Zhou, J., Hoi, S.: Mai-ui technical report: Real-world centric foundation gui agents (2025)
58. Zhou, S., Xu, F.F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al.: Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854 (2023)
59. Zhou, Y., Liu, L., Gou, C.: Learning from observer gaze:zero-shot attention prediction oriented by human-object interaction recognition. In: CVPR (2024)
60. Zhou, Y., Dai, S., Wang, S., Zhou, K., Jia, Q., Xu, J.: Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. arXiv preprint arXiv:2505.15810 (2025)
61. Zhu, X., Xia, M., Wei, Z., Chen, W.L., Chen, D., Meng, Y.: The surprising effectiveness of negative reinforcement in llm reasoning. arXiv preprint arXiv:2506.01347 (2025)
62. Zuo, Y., Zhang, K., Sheng, L., Qu, S., Cui, G., Zhu, X., Li, H., Zhang, Y., Long, X., Hua, E., et al.: Ttrl: Test-time reinforcement learning. arXiv preprint arXiv:2504.16084 (2025)