FDC-Ground: Improving GRPO for GUI Grounding via Exponential Rewards and Fact-Aligned Pruning

Xiangjian Zeng[1], **Wenjing Li**[2*], **Qingqiang Wu**[3,4,5,6,7], **Liang Zhang**[8]
[1]School of Journalism and Communication, Xiamen University
[2]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
[3]School of Film, Xiamen University
[4]School of Informatics, Xiamen University
[5]Xiamen Key Laboratory of Intelligent Storage and Computing, Xiamen University
[6]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of
Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen University
[7]Institute of Artificial Intelligence, Xiamen University
[8]Xiaohongshu Inc.
xjzeng@xmu.edu.cn, wjli@bupt.edu.cn, wuqq@xmu.edu.cn, zhangliang11@xiaohongshu.com

Abstract

This paper presents **FDC-Ground**, a reinforcement learning framework that addresses the high-cost, low-signal challenge of GUI grounding training. The framework introduces two core contributions: (1) the **Exponentially Decayed Distance Reward (EDDR)**, which provides resolution-robust and continuous feedback for position predictions, and (2) the **Fact-Aligned Dynamic Completions Pruning (FDC-Pruning)** strategy, which selectively retains completions whose advantage signs align with factual correctness, thereby reducing computational overhead while enhancing gradient quality and training stability. Using only 3.2K training samples and a single epoch, our 7B-parameter model achieves **88.3%** and **91.0%** accuracy on ScreenSpot and ScreenSpot-v2, outperforming several RL-based models such as UIShift and SE-GUI. Our 3B-parameter model based on Qwen2.5-VL-3B surpasses its original performance by **+26.6%**, demonstrating the effectiveness of our reward design and pruning strategy under low-resource conditions. Furthermore, the proposed FDC-Pruning strategy achieves a **1.18× training speedup** and a **+5.9% accuracy improvement** over standard GRPO, and expanding the exploration space to 4× yields an additional **+10.5%** gain, confirming both the scalability and the training efficiency of our approach. These findings highlight that combining EDDR with FDC-Pruning offers a practical path toward scalable and efficient RL-based GUI grounding, even in low-resource settings.

Code — https://github.com/mzengxj/FDC_Ground

Introduction

With the rise of large language models, web navigation and GUI agents have shown great potential in multimodal tasks (Zhang et al. 2025; Zheng et al. 2024; Kil et al. 2024; Wang et al. 2024b; Yang et al. 2024a). Recent work explores web environment modeling (Shi et al. 2017; Liu et al. 2018; Zhou et al. 2023), visual understanding (Bai et al. 2023; Liu et al. 2023; Bai et al. 2025), GUI grounding (Gou et al. 2024; Liu et al. 2025a; Lin et al. 2024), and reinforcement learning (Yuan et al. 2025; Luo et al. 2025). GUI grounding, which aligns GUI elements with language instructions, is a core capability in building GUI agents. (Hong et al. 2024; Cheng et al. 2024). While supervised fine-tuning (SFT) performs well (Gou et al. 2024; Xu et al. 2024; Wu et al. 2024), it relies on large-scale high-quality data and struggles to generalize to unseen interfaces.

Rule-based reinforcement fine-tuning (RFT) offers a scalable alternative by optimizing models with reward signals (Li et al. 2025), achieving competitive results with less data (Liu et al. 2025b; Shen et al. 2025; Huang et al. 2025). Group Relative Policy Optimization (GRPO) (Shao et al. 2024; Guo et al. 2025) is a lightweight RL method that avoids value functions.

GUI grounding typically requires selecting a single, precisely defined target element (e.g., a button), where all other predictions are considered incorrect. In such settings, reward functions often rely on binary rewards, 1 for correct selections and 0 for otherwise (Luo et al. 2025). However, this design leads to sparse learning signals when all predictions are incorrect, resulting in vanishing gradients. Designing unified reward functions across mobile, desktop and web (Lin et al. 2025a; Yang et al. 2024b; Kapoor et al. 2024) platforms further complicates training, especially in early stages with noisy predictions.

Early-stage exploration is critical in reinforcement learning (Yuan et al. 2022; Jiang, Kolter, and Raileanu 2023; Hao et al. 2025). However, GRPO's high computational cost under high-resolution settings limits the breadth of exploration, ultimately reducing training efficiency. Recent studies suggest that not all sampled solutions contribute positively to training in GRPO (Lin et al. 2025b), motivating our investigation into pruning strategies.

In this paper, we propose **FDC-Ground**, a reinforcement learning framework that improves the scalability of

*Corresponding author.

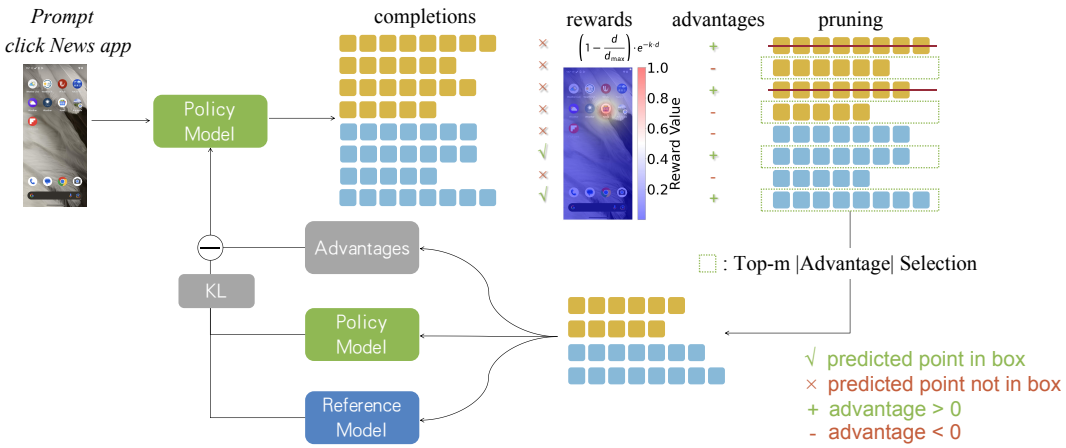

Figure 1: Overview of the FDC-Ground framework. The framework computes exponentially decayed distance rewards between predicted points and ground-truth box centers, then prunes completions with misleading advantages—such as predicted points that fall outside the box but still receive positive advantages—and updates the policy using the top-m samples. This enables dynamic expansion of the model's exploration space.

GRPO for GUI grounding. The framework introduces two key components: the **Exponentially Decayed Distance Reward (EDDR)**, and the **Fact-Aligned Dynamic Completions Pruning (FDC-Pruning)** strategy. EDDR provides continuous reward value across varying screen resolutions, and the closer a predicted point is to the ground-truth center, the higher the reward it receives. However, directly applying this reward reveals a new issue: many incorrect predictions (i.e., predicted points outside the ground-truth box) still receive positive advantages, leading to misleading gradient updates. To mitigate this, we introduce the FDC-Pruning strategy, which aligns gradient directions with factual correctness. This not only improves training stability but also enables broader exploration within constrained GPU budgets. Experiments on ScreenSpot and ScreenSpot-v2 validate the effectiveness of the proposed approach.

Related Work

GUI Grounding

In Graphical User Interface (GUI) agents, visual grounding is a core capability—the ability to accurately locate screen elements based on natural language instructions (Cheng et al. 2024; Zheng et al. 2024; Wang et al. 2024a). Early approaches relied on textual representations such as HTML code to extract page structure for grounding (Deng et al. 2023; Zhou et al. 2023). Recent work has explored large vision-language models (LVLMs) that combine vision encoders with LLMs to enable multimodal reasoning over image and text inputs (Gou et al. 2024; Bai et al. 2025).

Models like CogAgent(Hong et al. 2024), UGround(Gou et al. 2024), and OS-Atlas (Wu et al. 2024) focus on image-level element grounding via LVLMs, significantly improving grounding accuracy in scenarios without HTML access. However, these methods rely on supervised fine-tuning (SFT), which requires vast amount of diverse data and suffers from limited generalization. Therefore, it is necessary to explore more advanced learning paradigms to develop more capable and generalizable GUI agents.

Reinforcement Learning

Reinforcement learning (RL) has emerged as a promising alternative to supervised fine-tuning (SFT) in GUI grounding tasks (Yuan et al. 2025; Luo et al. 2025; Shen et al. 2025), where agents must interpret natural language instructions and take precise actions within complex visual interfaces. Unlike supervised fine-tuning (SFT), which relies heavily on large-scale annotated datasets and often suffers from poor generalization, reinforcement learning (RL) leverages rule-based reward functions to directly guide model behavior and improve performance. (Ye et al. 2025)

However, most existing reward functions are limited to binary rewards (0/1)(Luo et al. 2025) or squared decay functions(Yuan et al. 2025). Binary rewards suffer from sparsity and fail to provide useful training signals in the early stages. As shown in the Reward Design part of our method, squared decay functions also struggle to deliver stable feedback when training across varied screen resolutions.

Recent advances in reinforcement learning have introduced Group Relative Policy Optimization (GRPO) (Shao et al. 2024; Guo et al. 2025) as an effective approach for training GUI agents and reasoning models. GRPO eliminates the need for explicit value functions by comparing a group of sampled completions using rule-based rewards. However, its training efficiency is hindered by the need for multiple forward passes per prompt, as each completion must be evaluated independently. To address this limitation, CPPO (Lin et al. 2025b) proposes a pruning strategy that filters out low-quality completions based on their relative advantage, while CPPO significantly reduces GRPO's training cost by pruning completions with low absolute advantages, it uses discrete reward functions in mathematical reasoning tasks, where each completion is evaluated as either correct or incorrect based on final answers, Different

from GUI grounding settings where position predictions are evaluated approximately. However, if rewards are assigned to all predicted points generated by the model in the GUI grounding setting, some factually incorrect completions may still receive positive advantages and be mistakenly retained during pruning. And correct or near-correct completions with slightly lower advantage scores may be mistakenly discarded, weakening the gradient signal and hindering convergence.

Method

Policy Optimization for GUI Grounding

The GUI grounding task aims to locate a specific element on a graphical user interface (GUI) screenshot based on a natural language instruction. Formally, given a screenshot $s \in \mathbf{R}^{H\times W\times 3}$ and a textual instruction x, the model predicts a coordinate $\hat{y} \in \mathbf{R}^2$ or a bounding box $\hat{y} \in \mathbf{R}^4$ corresponding to the target element. The objective is to learn a model π_θ that predicts the target location $\hat{y}$ based on the pair (s, x), where x is the instruction prompt.

$$\hat{y} = \pi_\theta(s, x)$$

During training, the ground-truth bounding box y^* is provided, and a reward function $R(\hat{y}, y^*)$ is used to guide optimization. Specifically, we adopt a continuous reward signal that reflects spatial proximity between the predicted points and ground-truth positions, and further apply reinforcement learning to maximize expected reward:

$$\max_\theta \mathbf{E}_{(s,x,y^*)\sim\mathcal{D}}\left[R\left(\pi_\theta(s,x), y^*\right)\right]$$

GRPO is a reinforcement learning algorithm optimized for large-scale language models (LLMs) that has significant advantages in computing efficiency, training stability and applicability. Its original objective is defined as follows:

$$\begin{aligned}\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbf{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(o|q)} \Bigg\{ \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}|q,o_{i,<t})}A_i,\right. \\ \left.\mathrm{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}|q,o_{i,<t})},\,1-\epsilon,\,1+\epsilon\right)A_i\right] \\ -\beta\,\mathbf{D}_{\mathrm{KL}}\left[\pi_\theta\|\pi_{\mathrm{ref}}\right]\Bigg\}\end{aligned} \tag{1}$$

where π_θ is the current policy, $\pi_{\theta_{\mathrm{old}}}$ is the old policy, A_i denotes the advantage calculated based on relative rewards of the outputs, and π_{ref} is a reference policy.

In our image grounding task, we adopt a token-level loss inspired by DAPO (Yu et al. 2025), which provides finer-grained supervision. It helps the model better align visual and language features, suppress irrelevant or repetitive tokens, and produce more precise outputs. Following the training strategy of DeepSeekMath-RL, our policy model is updated once after each exploration stage. Consequently, we set $\pi_{\theta_{\mathrm{old}}} = \pi_\theta$, leading to the following token-level loss:

$$\begin{aligned}\mathcal{J}(\theta) = \mathbf{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(o|q)} \Bigg\{ \\ \frac{1}{G|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}|q,o_{i,<t})}A_i \\ -\beta\,\mathbf{D}_{\mathrm{KL}}\left[\pi_\theta\|\pi_{\mathrm{ref}}\right]\Bigg\}\end{aligned} \tag{2}$$

The model generates G candidate position predictions $\{o_1, o_2, \ldots, o_G\}$ for each instruction during training. Each output is evaluated by a reward function, producing rewards $\{r_1, r_2, \ldots, r_G\}$. And the relative advantage A_i is computed by normalizing each reward within the group as follows:

$$A_i = \frac{r_i - \mathrm{mean}(r)}{\mathrm{std}(r)}.$$

Additionally, we follow GRPO in applying a KL regularization term between the current policy π_θ and a reference policy π_{ref}, and optimize the overall objective by maximizing the token-level loss defined in Eq. (2).

Reward Design

In visual grounding, task success is typically determined by whether the predicted point falls within the target bounding box. However, using a binary reward function—assigning a reward of 1 when the prediction is inside the box or 0 otherwise—leads to two major problems:

- **Gradient Sparsity**: When predictions are incorrect, the binary reward provides no gradient signal, making it difficult for the model to learn effectively through backpropagation.
- **Reward Sparsity**: During the early training phase, randomly predicted point rarely fall inside the target box, resulting in mostly zero rewards across samples, which significantly reduces training efficiency.

To address these problems, we propose the **Exponentially Decayed Distance Reward (EDDR)** with the following key properties:

- **Continuity**: The reward value decays exponentially with the distance between the predicted point and the target, ensuring meaningful gradient signals even for incorrect predictions that outside the ground-truth bounding box.
- **Positive Reinforcement**: When the prediction falls within the correct region, an additional bonus is provided to accelerate convergence.
- **Center-Oriented Shaping**: A nonlinear mapping is applied to emphasize higher rewards near the center of the bounding box while reducing rewards at the edges, guiding the model toward more precise localization.

The proposed reward function is defined as follows:

$$R(p,g) = \begin{cases} \left(1 - \frac{d(p,c_g)}{d_{\max}}\right)\cdot e^{-k\cdot d(p,c_g)} + 0.5, & \text{if } p \in g \\ \left(1 - \frac{d(p,c_g)}{d_{\max}}\right)\cdot e^{-k\cdot d(p,c_g)}, & \text{otherwise} \end{cases}$$

where:

- p is the predicted point;

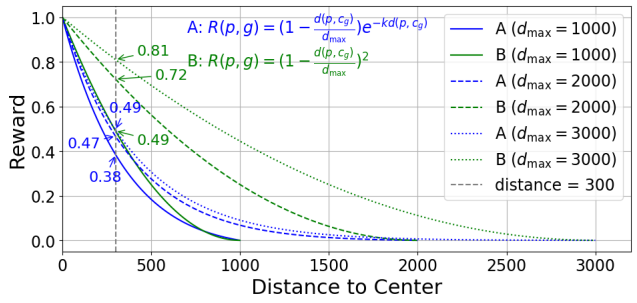

Figure 2: Comparison of the proposed reward function and a quadratic reward function at a fixed distance $d = 300$. Each curve simulates a different image resolution by varying the maximum distance $d_{\max} \in \{1000, 2000, 3000\}$. Our method maintains stable reward values across different image resolutions, while the quadratic reward increases significantly as resolution increases.

- g is the ground-truth bounding box;
- c_g denotes the center of g;
- $d(p, c_g)$ is the distance between p and c_g;
- $d_{\max}$ is the maximum distance in the image;
- k is the decay coefficient.

The distance $d(p, c_g)$ is defined as:

$$d(p, c_g) = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

where $p = (x_p, y_p)$ is the predicted point and $c_g = (x_c, y_c)$ is the center of the ground truth box g.
The maximum distance $d_{\max}$ is computed as the diagonal length of the image:

$$d_{\max} = \sqrt{\text{width}^2 + \text{height}^2}$$

where width and height are the image resolution.

In GUI grounding, image data often exhibit significant resolution variation due to differences across devices. To evaluate the robustness of our reward function across different image resolutions, we compare it with a quadratic reward function which has been used in prior work (Yuan et al. 2025). As shown in Figure 2, although the quadratic reflects distance changes, its reward values vary significantly with image resolution. For example, at a fixed distance of $d = 300$, the quadratic reward increases significantly from 0.49 to 0.81 as the maximum distance increases from $d_{\max} = 1000$ to $d_{\max} = 3000$. In contrast, our reward remains more consistent, changing only slightly from 0.21 to 0.27, demonstrating superior scale-invariant behavior.

Following previous work (Guo et al. 2025; Huang et al. 2025; Yuan et al. 2025), we also introduce format reward during training to check whether the generated output includes both the $< think >$ and $< answer >$ tags, ensuring that the response follows the expected structure. Finally, the reward consists of two components: (1) Format score, which evaluates the structural correctness of intermediate outputs; (2) Distance score, which measures the spatial accuracy of the predicted points.

Pruning Strategy

Gradient Decomposition. The gradient of the policy objective function in Eq. (2) with respect to the model parameters θ is given by:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbf{E}_{q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(o|q)} \Bigg\{ \frac{1}{G|o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \left[\hat{A}_{i,t} + \beta \left(\frac{\pi_{\text{ref}}(o_{i,t}|o_{i,<t})}{\pi_\theta(o_{i,t}|o_{i,<t})} - 1 \right) \right] \cdot \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \Bigg\} \tag{3}$$

The complete derivation is provided in Appendix A. This gradient can be decomposed into three terms: (1) the advantage term $\hat{A}_{i,t}$, indicating whether to increase or decrease the token's probability; (2) the KL penalty, which discourages large deviations from a reference policy; and (3) the log-probability gradient. Among them, the advantage term plays the most critical role, as it directly determines whether a completion should be reinforced or suppressed. However, its effectiveness hinges on whether high-advantage completions are indeed factually correct.

Exploration and Computational Cost. Exploration plays an indispensable role in the training of reinforcement learning agents, as it enables them to sample a diverse range of actions and uncover strategies that may not be immediately apparent (Hao et al. 2025). To quantify its effect in GUI grounding task, we adopt the *Hit@k* metric, which measures the proportion of prompts where at least one correct solution appears among k independently generated completions. As shown in Figure 3, Hit@k increases consistently with k for both Qwen2.5-VL-3B and Qwen2.5-VL-7B—reaching 73.13% and 77.75% at $k = 32$, respectively. While broader sampling improves the chance of identifying correct answers, it also leads to significant computational overhead, as GRPO requires a full forward computing for each candidate, especially costly in large-scale or high-resolution GUI tasks.

Fact-Aligned Pruning. Inspired by the pruning strategy adopted in CPPO, we propose a **Fact-Aligned Dynamic Completions Pruning (FDC-Pruning)** strategy. It performs **Dynamic** candidate generation until a correct answer appears or the maximum number of generations is reached, then filters the set of generated **Completions** by enforcing **Fact-aligned** consistency between advantage sign and correctness. Finally, **Pruning** is applied by selecting the top-m completions from the filtered set based on absolute advantage.

This design expands the exploration space while reducing redundant computation through dynamic completion generation. As revealed by the gradient analysis in Eq. (3), the advantage term dictates the optimization direction. However, CPPO's top-m pruning may select incorrect completions with positive advantages, leading to misguided updates (Appendix B). By enforcing fact-aligned filtering, FDC-Pruning ensures that policy updates are both accurate and efficient. The method proceeds as follows:

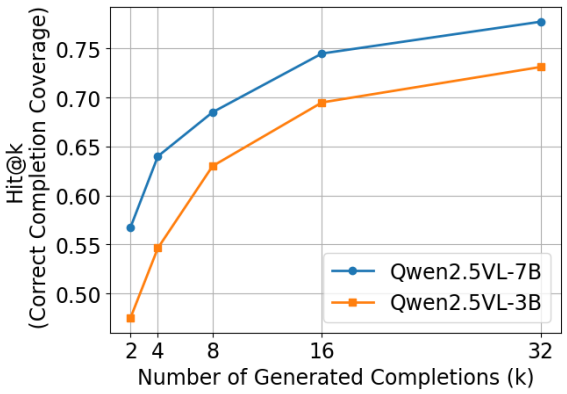

Figure 3: Hit@k correct completion coverage under varying numbers of generated completions per prompt. As k increases, the probability of sampling at least one correct answer also increases.

Given a set of n generated completions $\{o_i\}_{i=1}^n$ with corresponding rewards $\{r_i\}$ and advantages $\{\hat{A}_i\}$, we define the indicator of factual correctness as $y_i \in \{0, 1\}$, where $y_i = 1$ indicates a correct completion. In GUI grounding tasks, a completion is considered correct if its predicted point lies within the ground-truth bounding box.

We define the pruning set $\mathcal{S}$ as:

$$\mathcal{S} = \begin{cases} \textit{top-}m \text{ by } |\hat{A}_i| \text{ from } \mathcal{F}, & \text{if } \sum_{i=1}^n y_i \notin \{0, n\} \\ \textit{top-}m \text{ by } |\hat{A}_i|, & \text{otherwise} \end{cases}$$

where $\mathcal{F} = \left\{ i \mid (y_i = 1 \wedge \hat{A}_i > 0) \vee (y_i = 0 \wedge \hat{A}_i < 0) \right\}$ is the set of completions whose advantage signs align with their factual correctness.

Only completions in $\mathcal{S}$ are used for policy updates. A detailed procedural description of this filtering process is provided in Appendix I.

Experiments

Training Data Description

Our training data is constructed from two sources: the Aria-UI dataset (Yang et al. 2024b) for mobile and web data, and the ShowUI dataset (Lin et al. 2025a) for desktop data. We randomly sample 1,000 instances from each of Aria-UI Mobile and Aria-UI Web. For ShowUI, to enhance robustness on high-resolution screens, we apply a stratified sampling strategy: 1,000 samples from high-resolution screens (over 3,000 pixels wide) and 200 from lower-resolution ones. This sampling design balances training efficiency with resolution diversity, ensuring that the model is adequately exposed to high-resolution interfaces, which are typically underrepresented in existing datasets.

The resolution distribution across different device types is summarized in Appendix E.

Training Details

We adopt both Qwen2.5-VL-3B and Qwen2.5-VL-7B as our backbone models. The reward shaping coefficient k is set to 0.004. Due to hardware constraints, we limit the top-m selection values to 4.

During training, only the vision encoder and LLM layers are updated, while the MLP layers are kept frozen to reduce computational cost. We use DeepSpeed ZeRO-3 and FlashAttention-2 to improve training efficiency. All experiments are conducted on 8×NVIDIA A800 80GB GPUs for a single epoch. And use the same prompt across both training and evaluation stages.

Our implementation is based on the Open-R1-Multimodal[1] codebase, with minor modifications tailored to our GUI grounding task. The complete training configuration is provided in Appendix C.

Main Results

The evaluation results on the ScreenSpot and ScreenSpot-v2 benchmarks are summarized in Table 1. All experiments are conducted using a low generation temperature ($t = 0.1$), which ensures stable outputs from the language model during evaluation. Our FDC-Ground-7B model achieves an average accuracy of 88.3%, outperforming several models such as UIShift-7B (8 epochs, 87.8%) and InfiGUI-R1 (32K samples, 87.5%).Moreover, our model achieves state-of-the-art performance on specific sub-tasks, including 94.3% on desktop text and 91.7% on web text, highlighting its robust generalization ability across device types and modalities.

On the ScreenSpot-v2 benchmark, our method continues to demonstrate superior performance. As shown in Table 1, FDC-Ground-7B achieves an average accuracy of 91.0%, surpassing SE-GUI-7B (90.3%, 10 epochs) and matching the performance of UIShift-7B-2K (90.3%, 8 epochs), while requiring significantly fewer training steps. Notably, our model leads in Web-Text (95.7%) and Web-Icon (84.2%), two of the most visually complex and semantically rich subtasks, demonstrating its cross-modality reasoning capabilities under minimal supervision.

Our FDC-Ground-3B model, trained under the same training configuration (3.2K samples, 1 epoch), also exhibits strong performance. On ScreenSpot, our 3B model achieves an average accuracy of 82.1%, outperforming several larger SFT-based models such as CogAgent-18B (47.4%) and SeeClick-9.6B (53.4%), and approaching the performance of Aguvis-7B (84.4%). On ScreenSpot-v2, it reaches 85.5%, again surpassing most SFT models including OS-Atlas-7B (84.1%) and UGround-7B (76.3%).

Compared to the base vision-language model Qwen2.5-VL-3B, which achieves only 46.9% average accuracy on ScreenSpot-v2 despite being pretrained on large-scale data, our RL-trained FDC-Ground-3B model achieves a substantial +38.6% absolute improvement, underscoring the impact of our reward-driven optimization.

We also evaluate our method on 500 randomly sampled steps from the AndroidControl dataset (Li et al. 2024), following the UGround evaluation setting. As shown in Table 2, our model achieves strong performance in both high-level and low-level scenarios.

[1]https://github.com/EvolvingLMMs-Lab/open-r1-multimodal

Train Samples	Size	Epochs	Model	Mobile		Desktop		Web		**Spot**	**Spot-v2**
				Text	Icon	Text	Icon	Text	Icon	Avg	Avg
Close Models											
-	-	-	GPT-4o	30.5	23.2	20.6	19.4	11.1	7.8	18.8	-
-	-	-	Claude Computer Use	-	-	-	-	-	-	83.0	-
-	-	-	Gemini 2.0	-	-	-	-	-	-	84.0	-
Open-source Models											
-	3B	-	Qwen2.5-VL-3B	-	-	-	-	-	-	55.5	46.9
-	7B	-	Qwen2.5-VL-7B	-	-	-	-	-	-	84.7	86.5
SFT											
222M	18B	-	CogAgent-18B	67.0	24.0	74.2	20.0	70.4	28.6	47.4	-
1M	9.6B	-	SeeClick-9.6B	78.0	52.0	72.2	30.0	55.7	32.5	53.4	55.1
10M	7B	-	UGround-7B	82.8	60.3	82.5	63.6	80.4	70.4	73.3	76.3
13M	7B	-	OS-Atlas-7B	93.0	72.9	91.8	62.9	90.9	74.3	82.5	84.1
256K	2B	-	ShowUI-2B	92.3	75.5	76.3	61.1	81.7	63.6	75.1	77.3
1M	7B	-	Aguvis-7B	95.6	77.7	93.8	67.1	88.3	75.2	84.4	87.3
RL											
2K	7B	8	UIShift-7B-2K	96.7	85.2	91.2	75.7	89.6	82.0	87.8	90.3
3K	3B	9	GUI-R1-3B	-	-	93.8	64.8	89.6	72.1	-	-
3K	7B	9	GUI-R1-7B	-	-	91.8	73.6	91.3	75.7	-	-
32K	3B	-	InfiGUI-R1-3B	**97.1***	81.2	94.3	**77.1***	91.7	77.6	87.5	-
3K	7B	10	SE-GUI-7B	-	-	-	-	-	-	88.2	90.3
Ours											
3.2K	3B	1	FDC-Ground-3B	96.0	76.9	93.3	62.1	87.0	67.0	82.1	85.5
3.2K	7B	1	**FDC-Ground-7B**	96.3	**85.2***	**94.3***	73.6	**91.7***	**81.6***	**88.3***	**91.0***

Table 1: Accuracy (%) on ScreenSpot and ScreenSpot-v2. For ScreenSpot-v2, only average accuracy is reported, the full breakdown of ScreenSpot-v2 results is provided in Appendix D

Planner	**Grounding**	**High**	**Low**
GPT-4o	SeeClick	41.8	52.8
GPT-4o	UGround	48.4	62.4
GPT-4o	UGround-V1-2B	50.0	65.0
GPT-4o	UGround-V1-7B	49.8	66.2
GPT-4o	**FDC-Ground-7B**	**50.6**	**67.0**

Table 2: Step accuracy on AndroidControl over 500 random actions from the *UGround* (Gou et al. 2024) test dataset.

Group Size	**Web-Text**	**Web-Icon**	**Avg**	**Time (s)**
GRPO g=4	75.7	55.3	66.1	11541
4	80.4	62.6	72.0	9789
8	83.0	64.1	74.1	10926
16	**86.5**	65.5	**76.6**	19675
32	84.8	**66.5**	76.1	31665
64	82.1	66.5	74.8	57702

Table 3: Performance and training time under different group sizes on Qwen2.5-VL-3B (Web subset, $m = 2$).

Together, these findings reveal that our reinforcement learning framework effectively distills GUI understanding capabilities even under constrained data budgets, offering a competitive alternative to conventional supervised finetuning.

Impact of Group Size

Our method enables the model to expand its exploration space within a single iteration. We conducted a series of experiments on Qwen2.5-VL-3B to investigate how the exploration space affects model performance and training time. he model was trained on the Web subset of the training data and used a fixed top-m selection ($m = 2$) to enable pruning, while varying the group size $g \in \{4, 8, 16, 32, 64\}$. For comparison, we also include the GRPO baseline, setting $g = 4$. The results are shown in Table 3.

Compared to the GRPO baseline (no pruning), applying our **FDC-Pruning strategy** with $g = 4, m = 2$ achieves an improvement of +5.9% (from 66.1% to 72.0%) while reducing training time by 15.2% (acceleration factor of **1.18**×). The performance continues to improve with increasing group size up to $g = 16$, but then saturates or slightly declines as g grows larger.

To better understand this behavior, we analyze the moving average of the proportion of positive samples selected in the top-m completions during training. As shown in Figure 4, smaller group sizes (e.g., $g = 8$ or $g = 16$) maintain a higher and more stable rate of selecting correct samples, while larger groups suffer from a noticeable drop in this ratio during later training stages. This degradation stems from the nature of GRPO. When the group reward distribution becomes highly concentrated, low-reward samples (e.g., those with reward 0) may be assigned abnormally large advantage values. In late-stage training, where correct completions are common, such localized normalization amplifies noisy outliers, leading to erroneous advantage estimates and unstable policy updates.

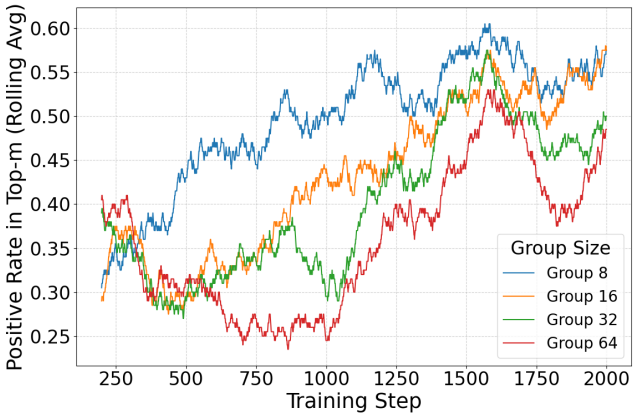

Figure 4: Rolling average of the proportion of positive samples in Top-m under different group sizes.

Setting	Mobile	Desktop	Web	Avg
Sparse Binary Reward + GRPO	86.7	72.8	61.7	74.4
Square Decayed Distance Reward + GRPO	89.4	83.5	82.6	85.5
EDDR + GRPO	89.6	82.0	84.6	85.9
EDDR + Top-m Selection	90.0	82.3	85.8	86.7
EDDR + FDC-Pruning (FDC-Ground)	**91.0**	**86.2**	**86.2**	**88.1**

Table 4: Ablation study on reward functions and training strategies. All results are accuracy (%) on ScreenSpot.

These findings suggest that moderate group sizes—approximately 4–8 times m—strike a favorable balance between exploration and stability. When combined with FDC-pruning strategy, such configurations substantially improve both accuracy and efficiency under limited supervision.

Ablation Study

To evaluate the individual and joint impact of our proposed Exponentially Decayed Distance Reward (EDDR) and the Fact-Aligned Dynamic Completions Pruning (FDC-Pruning) strategy, we conduct ablation experiments under five settings: sparse binary reward with GRPO, square decayed distance reward with GRPO, EDDR with GRPO, EDDR with Top-m selection which only prunes based on the absolute value of the advantage, and EDDR with FDC-Pruning.

Reward Function. Compared to the sparse binary reward and square decayed distance reward, our proposed EDDR improves performance from 74.4% to 85.9% and from 85.5% to 85.9%, respectively. We attribute this gain to the exponential shaping, which provides sharper gradients near the target and more consistent behavior across varying resolutions.

Training Strategy. Introducing Top-m selection with EDDR yields an additional gain of +0.8%. Finally, our full method—EDDR combined with Fact-Aligned Dynamic Completions Pruning (FDC-Pruning)—achieves 88.1% accuracy, outperforming the GRPO baseline by +2.6%. Both Top-m and FDC-Pruning experiments are conducted with a group size of $g = 16$ and a selection size of $m = 4$.

The results confirm that our reward design and pruning strategy jointly contribute to stable and efficient reinforcement learning for the GUI grounding task.

Conclusions and Limitations

Conclusions

To apply reinforcement learning to GUI grounding, we design an Exponentially Decayed Distance Reward, which assigns continuous rewards to all predicted points and increases rapidly near the center of the target box. However, many incorrect predictions—i.e., points outside the box—can still receive positive advantages, which may lead to incorrect optimization.

Furthermore, our experiments show that expanding the model's exploration space can improve performance. To balance effectiveness and efficiency, we propose a Fact-Aligned Pruning strategy based on the reward design. This strategy not only reduces computation but also allows broader exploration by focusing on completions whose advantage signs align with their factual correctness.

Experiments show that our method consistently outperforms standard GRPO and other strong RL baselines. We hope this work offers useful insights for future research on RL-based GUI grounding.

Limitations and Future Work

While reinforcement learning proves effective for fine-tuning large vision-language models, we observe that the reward signal typically increases during the first half of an epoch and then plateaus. This saturation limits further policy improvement. Recent studies (Cui et al. 2025; Cheng et al. 2025) suggest that entropy collapse may contribute to this stagnation. Although we experimented with entropy regularization techniques during training, no consistent improvement was observed. We leave more effective entropy control in the future work.

Second, we attempted to encourage broader exploration in the image space by designing exploration-oriented rewards. However, our experiments indicate that such reward functions tend to degrade model performance. Moreover, we observed that output length has little to no correlation with task accuracy in GUI grounding scenarios.

Finally, due to limited training data and computational resources, our experiments were constrained to moderate-resolution screens. Investigating model behavior in high-resolution GUI environments and designing more resolution-robust strategies remain important directions for future work.

Acknowledgements

This work was supported by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2024-1-01).

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Cheng, D.; Huang, S.; Zhu, X.; Dai, B.; Zhao, W. X.; Zhang, Z.; and Wei, F. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

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 preprint arXiv:2401.10935*.

Cui, G.; Zhang, Y.; Chen, J.; Yuan, L.; Wang, Z.; Zuo, Y.; Li, H.; Fan, Y.; Chen, H.; Chen, W.; et al. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36: 28091–28114.

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 preprint arXiv:2410.05243*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hao, Q.; Song, Y.; Liao, Q.; Yuan, J.; and Li, Y. 2025. Llm-explorer: A plug-in reinforcement learning policy exploration enhancement driven by large language models. *arXiv preprint arXiv:2505.15293*.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14281–14290.

Huang, W.; Jia, B.; Zhai, Z.; Cao, S.; Ye, Z.; Zhao, F.; Xu, Z.; Hu, Y.; and Lin, S. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Jiang, Y.; Kolter, J. Z.; and Raileanu, R. 2023. On the importance of exploration for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 12951–12986.

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. In *European Conference on Computer Vision*, 161–178. Springer.

Kil, J.; Song, C. H.; Zheng, B.; Deng, X.; Su, Y.; and Chao, W.-L. 2024. Dual-view visual contextualization for web navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14445–14454.

Li, M.; Zhong, J.; Zhao, S.; Lai, Y.; Zhang, H.; Zhu, W. B.; and Zhang, K. 2025. Think or not think: A study of explicit thinking in rule-based visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.16188*.

Li, W.; Bishop, W. E.; Li, A.; Rawles, C.; Campbell-Ajala, F.; Tyamagundlu, D.; and Riva, O. 2024. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37: 92130–92154.

Lin, K. Q.; Li, L.; Gao, D.; Yang, Z.; Bai, Z.; Lei, W.; Wang, L.; and Shou, M. Z. 2024. Showui: One vision-language-action model for generalist gui agent. In *NeurIPS 2024 Workshop on Open-World Agents*, volume 1.

Lin, K. Q.; Li, L.; Gao, D.; Yang, Z.; Wu, S.; Bai, Z.; Lei, S. W.; Wang, L.; and Shou, M. Z. 2025a. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 19498–19508.

Lin, Z.; Lin, M.; Xie, Y.; and Ji, R. 2025b. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*.

Liu, E. Z.; Guu, K.; Pasupat, P.; Shi, T.; and Liang, P. 2018. Reinforcement Learning on Web Interfaces using Workflow-Guided Exploration. In *International Conference on Learning Representations (ICLR)*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36: 34892–34916.

Liu, Y.; Li, P.; Xie, C.; Hu, X.; Han, X.; Zhang, S.; Yang, H.; and Wu, F. 2025a. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*.

Liu, Z.; Sun, Z.; Zang, Y.; Dong, X.; Cao, Y.; Duan, H.; Lin, D.; and Wang, J. 2025b. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*.

Luo, R.; Wang, L.; He, W.; and Xia, X. 2025. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shen, H.; Liu, P.; Li, J.; Fang, C.; Ma, Y.; Liao, J.; Shen, Q.; Zhang, Z.; Zhao, K.; Zhang, Q.; et al. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Shi, T.; Karpathy, A.; Fan, L.; Hernandez, J.; and Liang, P. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, 3135–3144. PMLR.

Wang, J.; Xu, H.; Ye, J.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024a. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P. P.; et al. 2024. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; and Xiong, C. 2024. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.

Yang, K.; Liu, Y.; Chaudhary, S.; Fakoor, R.; Chaudhari, P.; Karypis, G.; and Rangwala, H. 2024a. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*.

Yang, Y.; Wang, Y.; Li, D.; Luo, Z.; Chen, B.; Huang, C.; and Li, J. 2024b. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*.

Ye, Y.; Huang, Z.; Xiao, Y.; Chern, E.; Xia, S.; and Liu, P. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuan, M.; Li, B.; Jin, X.; and Zeng, W. 2022. Rewarding episodic visitation discrepancy for exploration in reinforcement learning. *arXiv preprint arXiv:2209.08842*.

Yuan, X.; Zhang, J.; Li, K.; Cai, Z.; Yao, L.; Chen, J.; Wang, E.; Hou, Q.; Chen, J.; Jiang, P.-T.; et al. 2025. Enhancing Visual Grounding for GUI Agents via Self-Evolutionary Reinforcement Learning. *arXiv preprint arXiv:2505.12370*.

Zhang, C.; Yang, Z.; Liu, J.; Li, Y.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–20.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.