GROUNDGUI: GUI GROUNDING BY HIGH-QUALITY DATA AND DISTILLATION-TO-REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Constructing robust GUI agents relies on multimodal visual grounding—the ability to precisely map natural language intents to actionable on-screen components. Despite abundant desktop, web, and mobile benchmarks, high-fidelity resources tailored for diverse environments are surprisingly scarce. To bridge this gap, we introduce the Screen Referring Grounding Dataset (ScreenRef), a comprehensive GUI grounding dataset consisting of GUI screenshots paired with natural language referring expressions and box or point annotations. Spanning desktop, web, and mobile domains, ScreenRef comprises 586K screenshots and over 633K referring expression instances, covering tasks such as text recognition, visual matching, spatial understanding, and refusal scenarios. To boost the model's grounding performance, we propose the Distillation-to-Reinforcement Learning (D2RL) method to address the limited generalization of supervised fine-tuning and the sparse reward problem in reinforcement learning. Utilizing ScreenRef, we train Qwen3-VL-8B-Instruct via sequential SFT and D2RL training, yielding GroundGUI-8B. GroundGUI-8B achieves 95.2% on ScreenSpot-V2, 63.9% on ScreenSpot-Pro, 65.4% on OSWorld-G, 86.8% on MMBench-GUI L2, and 38.3% on UI-Vision. Among 8B-scale models, it produces substantial gains over models trained on open-source datasets, with performance comparable or even superior to models trained on closed-source data.

1 INTRODUCTION

Graphical User Interfaces (GUIs) (Cheng et al., 2024) are ubiquitous in modern human-computer and mobile interactions. Visual GUI grounding (mapping language instructions to pixel coordinates of interactive elements) is a foundational capability. Robust cross-platform grounding is challenging due to interface complexity and contextual diversity. Modern applications often have high-resolution canvases with dense layouts and subtle visual disambiguations, complicating spatial localization. GUI agents degrade on domain-specific software due to limited grasp of specialized icon semantics. Data underpins model capability, but constructing high-quality grounding datasets with broad coverage is challenging (Xu et al., 2024; Xie et al., 2025). Mobile-Agent-v3.5 (Xu et al., 2026), UI-Venus-1.5 (Gao et al., 2026), UI-TARS-2 (Wang et al., 2025a), and MAI-UI (Zhou et al., 2025) perform well with proprietary large-scale data, unavailable to the community. GroundCUA (Feizi et al., 2026) and ScaleCUA (Liu et al., 2026b) release their data, but models trained on them are suboptimal due to limited diversity/accuracy.

We introduce ScreenRef, a large-scale GUI grounding dataset (586K screenshots and 633K referring expressions), with text recognition, visual matching, spatial understanding, and refusal tasks. It covers desktop, web, and mobile platforms, with image resolutions ranging from 0.1 to 8.3M pixels. Multi-stage filtering/verification yield verified data for robust cross-platform grounding.

Supervised fine-tuning (SFT) is commonly used to train grounding models, but overfits to specific domains (Luo et al., 2025). Reinforcement learning (RL) works (Tang et al., 2026; Li et al., 2026; Liu et al., 2026a; Lu et al., 2026) use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to improve generalization, but suffer from sparse signals on hard samples where all rollouts receive zero reward. Since on-policy self-distillation (OPSD) (Zhao et al., 2026a) provides dense

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

token-level supervision, we propose **Distillation-to-Reinforcement Learning (D2RL)**, a two-stage post-training strategy: OPSD followed by REINFORCE Leave-One-Out (RLOO) (Ahmadian et al., 2024). OPSD tailored for GUI grounding delivers rich token-level supervision for precise coordinate generation from a teacher that sees the ground-truth bounding box, providing a stronger starting policy before RL. RLOO then enhances model generalization and localization accuracy. SFT and D2RL on the proposed data yield GroundGUI for precise GUI grounding across desktop, web, and mobile. GroundGUI-8B achieves 95.2% on ScreenSpot-V2, 63.9% on ScreenSpot-Pro, 65.4% on OSWorld-G, 86.8% on MMBench-GUI L2, and 38.3% on UI-Vision, remarkably outperforming the models trained on open-source datasets. GroundGUI-8B also surpasses Holo2-8B (H Company, 2025) and Step-GUI-8B (Yan et al., 2025) by a large margin, both of which are trained on closed-source data.

Core contributions. (1) **ScreenRef**, a high-quality GUI grounding dataset with 586K screenshots and 633K referring expressions, providing text recognition, visual matching, spatial understanding, and refusal supervision for robust GUI agents. We will release data, code, and models. (2) **D2RL**, which proposes OPSD before RL to shape a better initial policy, unleashing the potential of RL for GUI grounding, enhancing model generalization and grounding accuracy. (3) **GroundGUI-8B**, trained on our proposed data and approach. GroundGUI-8B achieves state-of-the-art or competitive performance among models with 8B parameters or fewer, surpassing several 32B-scale models, e.g., ScaleCUA-32B, OpenCUA-32B and GTA1-32B.

2 Related Works

GUI Agent. Computer-use agents interact with interfaces, from simple commands to multi-step tasks. CogAgent (Hong et al., 2024), ShowUI (Lin et al., 2025), Ferret-UI (You et al., 2024), OS-Genesis (Sun et al., 2025), GUI-Actor (Wu et al., 2025a), and UI-TARS (Qin et al., 2025) improve interaction via SFT. OS-ATLAS (Wu et al., 2025b) and UGround (Gou et al., 2025) scale data to map language to UI components. These methods are data inefficient, particularly in complex desktop and web environments. RL approaches, inspired by GRPO (Shao et al., 2024) (GUI-R1 (Luo et al., 2025), GUI-G^2 (Tang et al., 2026), GUI-C^2 (Li et al., 2026), InfiGUI-G1 (Liu et al., 2026a)), improve grounding. Still, reliably grounding instructions to on-screen elements remains a bottleneck. We improve GUI grounding with high-quality, diverse data and SFT followed by D2RL.

GUI Grounding Datasets cover mobile, web, and desktop, each with trade-offs. Mobile datasets (e.g., RICO (Deka et al., 2017), UIBert (Bai et al., 2021), AITW (Rawles et al., 2023), AITZ (Zhang et al., 2024), AMEX (Chai et al., 2025)) provide element-level supervision in standardized layouts, which simplifies extraction but offers limited layout density and visual variation vs. desktop software. Web-focused datasets (e.g., SeeClick (Cheng et al., 2024), WebUI (Wu et al., 2023), UGround, and AGUVIS-G (Xu et al., 2024)) scale collection by harvesting HTML/DOM structures. These favor text-bearing elements over icon-only controls.

High-quality desktop datasets remain scarce and hard to curate. OS-ATLAS extracts desktop elements from accessibility trees; inconsistent accessibility APIs introduce label noise. JEDI (Xie et al., 2025) scales dataset size via synthetic interface generation, but its simplified screens fail to capture desktop complexity. Recent efforts such as GroundCUA (Feizi et al., 2026), ScaleCUA (Liu et al., 2026b), and Wild2GUI (Xiong et al., 2026) offer dense annotations, cross-platform coverage, or video-synthesized data, but annotation inaccuracies or sparse diversity constrain performance.

3 ScreenRef Dataset

3.1 Screenshot Crawling

We use platform-specific approaches for GUI metadata extraction (Figure 1, Path 1). For desktop, Accessibility Trees (A11y Trees) provide interface layouts and UI element coordinates. Document Object Model (DOM) structures offer element information for web browsers. Parsed XML layout files provide UI element locations for Android applications. Metadata contains bounding boxes, categories, and text. However, some metadata contains noisy bounding box attributes. For these cases, we adopt OmniParser V2 (Yu et al., 2026) to reduce noise by comparing the metadata bounding boxes with the detection results of OmniParser. The bounding boxes with an IoU lower than 0.5

Table 1: Comparison with existing GUI grounding datasets. Instruction counts follow each dataset's original reporting unit and are not directly comparable across datasets; details in Appendix B.

	Platform			Scale & Statistics			
Dataset	**Web**	**Mobile**	**Desktop**	**Images**	**Instr.**	**Res Range**	**Refusal**
OS-ATLAS (Wu et al., 2025b)	✓	✓	✓	1.85M	—	(0.5, 5.2)	×
FineWeb (Penedo et al., 2024)	✓	×	×	1.4M	—	(2.1, 2.1)	×
UGround (Gou et al., 2025)	✓	×	×	773K	—	(0.4, 1.9)	×
Aria-UI (Yang et al., 2025)	✓	✓	✓	295K	—	(1.3, 1.9)	×
SeeClick (Cheng et al., 2024)	✓	×	×	270K	—	(2.1, 2.1)	×
GroundCUA (Feizi et al., 2026)	×	×	✓	55K	700K	(0.4, 7.0)	×
AMEX (Chai et al., 2025)	×	✓	×	101K	—	(0.9, 4.5)	×
JEDI (Xie et al., 2025)	✓	✓	✓	575K	4M	(0.9, 2.1)	✓
MolmoPoint-GUISyn (Clark et al., 2026)	✓	✓	✓	36K	10M	(0.9, 8.3)	×
AGUVIS-G (Xu et al., 2024)	✓	✓	✓	452K	1.04M	(0.5, 2.1)	×
UIBert (Bai et al., 2021)	×	✓	×	57K	—	(0.5, 2.1)	×
RICOSCA (Li et al., 2020a)	×	✓	×	18K	—	(0.5, 2.1)	×
Widget Caption (Li et al., 2020b)	×	✓	×	14K	—	(0.5, 2.1)	×
SCREENREF (ours)	✓	✓	✓	586K	633K	**(0.1, 8.3)**	✓

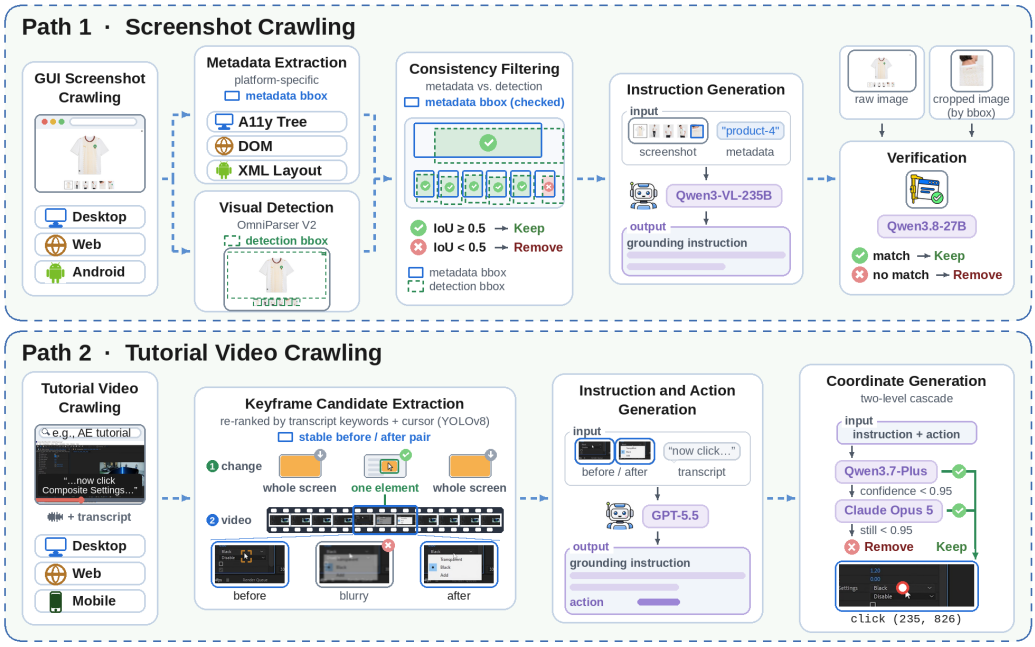

Figure 1: **Overview of the ScreenRef data construction pipeline. Path 1** consists of screenshot crawling, metadata filtering, instruction generation, and verification stages. **Path 2** contains tutorial video crawling, keyframe candidate extraction, joint instruction and action generation, and coordinate generation stages.

are removed. Figure 6 in Appendix D shows the annotation for the screenshot. Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025) is then employed to produce grounding instructions with annotated bounding boxes, categories, and surrounding context. To further verify the correctness of the generated grounding pairs, we adopt Qwen3.8-27B (Qwen Team, 2026c) to judge whether the answer matches the instruction given the raw image and the bounding box crop. Through this pipeline, we generate three primary types of instructions, including text recognition, visual matching, and spatial understanding. Text recognition denotes grounding a target element uniquely designated by readable on-screen text. Visual matching refers to identifying the target element by its visual attributes (icon noun, shape, or color). Spatial understanding represents identifying a target element purely through its spatial relationship, whether relative to an anchor element or bounded within a named container. Qualitative examples are provided in Appendix A.

3.2 TUTORIAL VIDEO CRAWLING

To collect real-world interfaces, we complement screenshots with interaction frames from public tutorial videos. To this end, search queries are automatically generated to crawl video metadata on YouTube. Qwen3.6-27B (Qwen Team, 2026a) is then used to filter out non-tutorial video metadata, followed by video downloading based on the preserved metadata. For subtitle-free videos, FFmpeg is exploited to extract audio for transcript generation. To produce grounding samples, we design a pipeline converting tutorial videos into trainable (instruction, observation, action, coordinate) tuples via 3 sequential stages: (i) **Keyframe Candidate Extraction.** It computes block-level frame differences to pinpoint timestamps of local screen updates. To prevent global screen transitions (shot cuts, page scrolling, video playback) from dominating, we down-weight full-frame edits, letting fine-grained interactions compete. Candidates are re-ranked using action verbs extracted from the transcript alongside cursor coordinates detected via YOLOv8 (Reis et al., 2023) as cheap priors, yielding at most 60 filtered candidates/video. Instead of capturing the immediate moment of change (often with motion blur or mid-animation states), we perform a forward-and-backward search around each timestamp to identify the nearest static frames, constructing a clean pre-/post-change frame pair. (ii) **Joint Instruction and Action Annotation.** Each candidate frame pair, with absolute timestamps, is concatenated with its transcript segment and video metadata into a single GPT-5.5 API prompt. This pass performs both task segmentation and action annotation, outputting a sequence of temporal actions. Each action includes its interaction type, parameter values, and an element-level referring expression grounded strictly in visible screen evidence (referencing UI controls solely by their visible text labels, icon shapes, or explicit functional descriptions). (iii) **Post-Hoc Visual Grounding.** Taking the pre-action frame as the visual observation, the target coordinate is resolved via action-type-conditioned routing. A single call to Qwen3.7-Plus (Qwen Team, 2026b) predicts target presence, a bounding box/coordinate, and a confidence score. If confidence falls below a threshold $\tau = 0.95$, the proposal is escalated to Claude Opus 5 for re-grounding. Samples with secondary confidence below τ are discarded. To fill the critical gap of handling infeasible grounding queries in existing datasets, we construct refusal samples via strategically random-pairing instructions and frames. We then employ the Qwen3.8-27B model to identify and verify genuinely infeasible grounding instances.

3.3 DATASET STATISTICS

Through dual-mode pipelines, we obtain ScreenRef, comprising 585,981 screenshots and 633,369 instructions (Instr), with 284,531 grounding samples from pathway 1 and the remaining from pathway 2. As listed in Table 1, ScreenRef features multiple platforms and a wider resolution range (Res Range) than the existing datasets. ScreenRef and JEDI (Xie et al., 2025) are the only two datasets that include refusal samples. The distribution of the 4 task types across desktop, web, and mobile environments is given in Table 9.

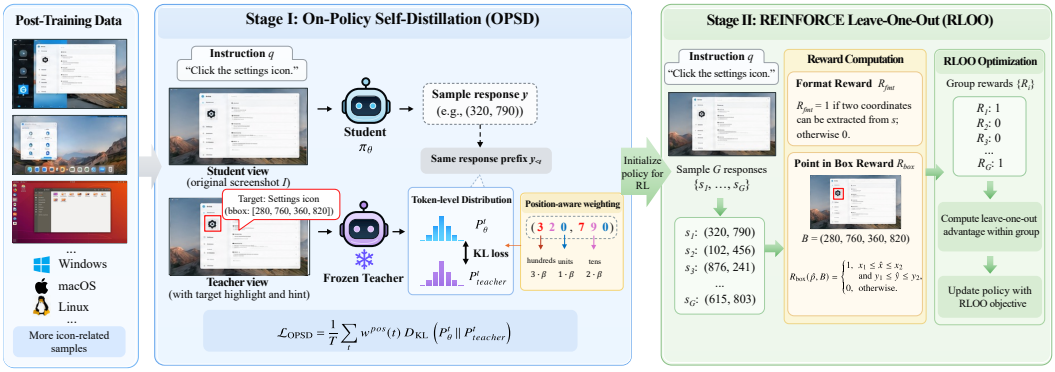

Figure 2: **Overview of the D2RL method.** The format check is a gate, not a reward term.

4 DISTILLATION-TO-REINFORCEMENT LEARNING

SFT first fits the model to ScreenRef, improving grounding across diverse environments. RL then directly optimizes the task-level click reward. However, RL strongly depends on its starting policy. On challenging samples such as small icons or spatial-relation queries, all sampled rollouts may fail and receive zero reward, providing little learning signal. We thus propose **Distillation-to-Reinforcement Learning (D2RL)**, which applies on-policy self-distillation (OPSD) (Zhao et al., 2026a) as an explicit preparation stage: dense teacher supervision on the student's own trajectories shapes a better starting policy before RL optimizes grounding success under the original screenshot (Table 8 confirms the benefit of the stronger initial policy provided by OPSD).

Stage I: On-Policy Self-Distillation. SFT learns from labeled responses, while RL samples from the model's own policy. OPSD bridges the two by matching the teacher's token distribution on student-generated prefixes via reverse KL divergence, providing guidance on the responses the model currently explores. Specifically, given a screenshot I and an instruction q, the student policy π_θ samples a coordinate response s from the original screenshot. Inspired by GUI-SD (Zhang et al., 2026), the teacher is a frozen copy of the SFT policy; during OPSD only, it receives the screenshot with the ground-truth target box highlighted together with a textual localization hint (the target name and its box), whereas the student receives only the original screenshot and the instruction. The teacher is not used at inference. Both models score the same student-sampled prefix, so the teacher provides dense token-level guidance along the student's rollout. Since errors in higher-order coordinate digits induce much larger spatial deviations than errors in lower-order digits, we apply position-aware weighting during distillation: we set $w^{\mathrm{pos}}(t) = \beta \cdot k_t$ for coordinate-digit tokens and $w^{\mathrm{pos}}(t) = 1$ otherwise; k_t denotes the significance of the digit (e.g., 1 for units, 2 for tens, and 3 for hundreds), and β controls the weighting strength. The OPSD objective is then

$$\mathcal{L}_{\mathrm{OPSD}} = \frac{1}{T} \sum_t w^{\mathrm{pos}}(t) D_{\mathrm{KL}} \left(P_\theta^t \,\|\, P_{\mathrm{teacher}}^t \right), \tag{1}$$

where P_θ^t and P_{teacher}^t denote the student and teacher token distributions at decoding step t. Stage I aims to improve the RLOO starting policy, not only pre-RL accuracy. Dense privileged supervision shifts the student's trajectories toward target-consistent coordinates, reducing sparse reward reliance. Position-aware weighting emphasizes coordinate decisions that matter for localization.

Stage II: REINFORCE Leave-One-Out. Starting from the OPSD-trained policy, we further optimize the grounding policy with the RLOO method (Ahmadian et al., 2024) under the original screenshot observations. Given a screenshot I and an instruction q, the policy π_θ generates a coordinate response s. Each response is evaluated using a simple point-in-box reward. If the response cannot be parsed into a click point, it receives a reward of 0; the format check in Figure 2 thus acts as a gate rather than an additive reward term. Otherwise, the parsed point $\hat{\mathbf{p}} = (\hat{x}, \hat{y})$ is evaluated in the normalized coordinate space. Let $B = (x_1, y_1, x_2, y_2)$ denote the target box, where (x_1, y_1) and (x_2, y_2) are its top left and bottom right corners. The point-in-box reward is

$$R_{\mathrm{box}}(\hat{\mathbf{p}}, B) = \begin{cases} 1, & x_1 \le \hat{x} \le x_2 \text{ and } y_1 \le \hat{y} \le y_2, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

For each screenshot and instruction pair, we sample G responses $\{s_i\}_{i=1}^G$. Following RLOO, the reward of each rollout is compared with the average reward of the other rollouts in the same group, providing a leave-one-out baseline without a separate critic. Let R_i denote the (point-in-box) reward of s_i. The policy gradient is given by

$$\nabla_\theta J(\pi_\theta) = \frac{1}{G} \sum_{i=1}^{G} \left(R_i - \frac{1}{G-1} \sum_{j \neq i} R_j \right) \nabla_\theta \log \pi_\theta(s_i \mid I, q). \tag{3}$$

The 2 stages provide complementary feedback. OPSD uses privileged information to guide student token distribution, including failed rollouts with identical click rewards, and prepares a better starting policy for RLOO, which uses task-level success to reweight responses under the original screenshot.

Table 2: Performance on **ScreenSpot-V2**. **Bold** values = best results.

Model	Mobile		Desktop		Web		Avg.
	Text	Icon.	Text	Icon.	Text	Icon.	
Open-Source Models							
Qwen3-VL-2B (Bai et al., 2025)	95.5	82.0	95.4	73.6	89.7	76.4	86.7
Phi-Ground-7B (Zhang et al., 2025)	90.2	76.4	93.6	75.9	96.5	62.0	83.8
OS-ATLAS-7B (Wu et al., 2025b)	95.2	75.8	90.7	63.6	90.6	77.3	85.1
UGround-V1-7B (Gou et al., 2025)	83.6	90.5	85.8	86.3	95.5	83.2	87.7
UI-TARS-1.5-7B (Qin et al., 2025)	92.2	81.5	91.0	84.2	95.5	84.5	89.0
SE-GUI-7B (Yuan et al., 2025)	99.3	89.1	96.4	78.6	92.7	81.3	90.8
UI-TARS-7B (Qin et al., 2025)	96.9	89.1	95.4	85.0	93.6	85.2	91.6
Qwen3-VL-8B (Bai et al., 2025)	97.9	84.8	95.9	87.9	95.7	83.7	91.7
GUI-Actor-7B (Wu et al., 2025a)	97.6	88.2	96.9	85.7	93.2	86.7	92.1
OpenCUA-7B (Wang et al., 2025c)	–	–	–	–	–	–	92.3
GTA1-7B (Yang et al., 2026)	99.0	88.6	94.9	89.3	92.3	86.7	92.4
GUI-Owl-7B (Wang et al., 2025b)	99.0	92.4	96.9	85.0	93.6	85.2	92.8
GUI-G^2-7B (Tang et al., 2026)	98.3	91.9	95.4	89.3	94.0	87.7	93.3
InfiGUI-G1-7B (Liu et al., 2026a)	99.0	91.9	94.3	82.1	**97.9**	89.2	93.5
UI-Venus-7B (Gu et al., 2025)	99.0	90.0	96.9	90.7	96.2	88.7	94.1
MAI-UI-8B (Zhou et al., 2025)	99.3	89.1	99.0	92.1	**97.9**	91.1	95.2
GUI-Owl-1.5-8B (Xu et al., 2026)	97.4	90.5	96.4	90.7	94.2	89.7	93.7
Holo2-8B (H Company, 2025)	–	–	–	–	–	–	93.2
MolmoPoint-8B (Clark et al., 2026)	–	–	–	–	–	–	93.4
Step-GUI-8B (Yan et al., 2025)	–	–	–	–	–	–	95.1
COPC-8B (Wang et al., 2026)	–	–	–	–	–	–	91.3
Qwen3-VL-32B (Bai et al., 2025)	96.2	90.0	97.4	85.0	95.7	89.7	93.0
GUI-Owl-32B (Wang et al., 2025b)	98.6	90.0	97.9	87.8	94.4	86.7	93.2
OpenCUA-32B (Wang et al., 2025c)	–	–	–	–	–	–	93.4
ScaleCUA-32B (Liu et al., 2026b)	98.6	91.9	99.0	90.0	94.4	91.6	94.7
GTA1-32B (Yang et al., 2026)	**99.7**	90.5	99.0	**94.3**	95.7	90.1	95.2
UI-Venus-72B (Gu et al., 2025)	**99.7**	**93.8**	95.9	90.0	96.2	**92.6**	**95.3**
UI-TARS-72B (Qin et al., 2025)	94.8	86.3	91.2	87.9	91.5	87.7	90.3
GroundGUI-8B	98.6	90.0	**99.5**	92.1	96.2	**92.6**	95.2

5 Experimental Results

5.1 Implementation Details

SFT is performed on 4 B300 GPUs, using a global batch size of 32 and a learning rate of 1×10^{-6}. The model is trained for 2 epochs using the AdamW (Loshchilov & Hutter, 2019) optimizer. In the second epoch, the loss weights for text recognition and refusal samples are reduced to 0.2. Starting from the SFT model, the model is post-trained with D2RL on 12K grounding samples for 2 epochs (one epoch for OPSD and one for RLOO) using 8 B300 GPUs. Both stages use AdamW with a learning rate of 1×10^{-6}, a cosine schedule, and a batch size of 42 prompts, without a KL penalty. OPSD matches a frozen teacher's token distributions on the policy's own rollouts, followed by RLOO with 8 rollouts per prompt and a binary point-in-box reward.

5.2 Evaluation Benchmarks

We evaluate GroundGUI on 5 representative GUI grounding benchmarks: ScreenSpot-V2 (SS-V2) (Wu et al., 2025b), MMBench-GUI L2 (MMB-GUI L2) (Wang et al., 2025d), ScreenSpot-Pro (SS-Pro) (Li et al., 2025), OSWorld-G (OSW-G) (Xie et al., 2025), and UI-Vision (Nayak et al., 2025). ScreenSpot-V2 evaluates general-purpose GUI grounding across web, desktop, and mobile, covering text, icons, and interactive elements. MMBench-GUI L2 evaluates more complex instructions requiring stronger interface understanding beyond basic element matching across multiple platforms. ScreenSpot-Pro focuses on high-resolution professional software, challenging models to localize small targets and distinguish densely arranged, visually similar UI elements. OSWorld-G derives from realistic computer-use scenarios and emphasizes layout understanding, operation semantics, and fine-grained interaction reasoning. UI-Vision evaluates basic, functional, and spatial

Table 3: Performance comparison on **MMBench-GUI L2** benchmark. **Bold** values = best results.

Model	Windows		macOS		Linux		iOS		Android		Web		Avg.
	Bas.	Adv.	Bas.	Adv.	Bas.	Adv.	Bas.	Adv.	Bas.	Adv.	Bas.	Adv.	
Qwen3-VL-2B (Bai et al., 2025)	81.9	0.0	80.3	46.2	67.5	0.0	90.8	0.0	91.0	0.0	88.1	0.0	46.5
OS-ATLAS-7B (Wu et al., 2025b)	36.9	18.8	44.4	21.7	31.4	13.3	74.8	48.8	69.6	46.8	61.3	35.4	41.4
Aguvis-7B (Xu et al., 2024)	37.3	21.7	48.1	33.3	33.5	25.0	67.5	65.2	61.0	51.0	61.6	45.5	45.7
UI-TARS-1.5-7B (Qin et al., 2025)	68.3	39.0	69.0	44.5	64.4	37.8	88.5	69.4	90.5	69.3	81.0	56.5	64.3
UGround-V1-7B (Gou et al., 2025)	66.8	39.0	71.3	48.6	56.5	31.1	92.7	70.9	93.5	71.0	88.7	64.6	65.7
GUI-Actor-7B (Wu et al., 2025a)	80.8	55.1	81.4	60.4	64.9	41.8	94.3	82.7	93.5	79.7	89.7	72.1	76.5
SE-GUI-7B (Yuan et al., 2025)	77.5	57.7	77.1	60.7	68.6	44.9	95.5	80.0	95.5	83.7	89.7	68.8	76.6
Qwen3-VL-8B (Bai et al., 2025)	88.6	61.8	85.5	69.1	74.9	53.1	95.2	82.4	95.5	84.5	**96.8**	72.1	81.3
GTA1-7B (Yang et al., 2026)	76.8	57.4	80.3	63.9	68.6	53.6	93.9	83.3	96.3	84.5	90.3	74.7	78.5
GUI-G^2-7B (Tang et al., 2026)	79.7	55.1	79.7	64.7	69.6	50.0	95.2	82.7	96.6	85.4	91.9	75.6	78.8
GUI-Owl-7B (Wang et al., 2025b)	86.4	61.8	81.7	64.5	74.4	61.7	94.9	83.0	95.8	83.7	93.2	72.7	80.5
InfiGUI-G1-7B (Liu et al., 2026a)	82.7	61.8	83.8	63.9	72.3	52.0	94.9	89.4	95.2	85.6	93.5	76.3	80.8
Step-GUI-8B (Yan et al., 2025)	-	-	-	-	-	-	-	-	-	-	-	-	85.6
COPC-8B (Wang et al., 2026)	-	-	-	-	-	-	-	-	-	-	-	-	78.4
GUI-Owl-32B (Wang et al., 2025b)	85.6	65.1	84.9	67.1	77.0	**63.3**	95.2	85.5	96.1	87.0	95.5	80.8	83.0
GTA1-32B (Yang et al., 2026)	82.3	66.9	89.0	74.0	73.3	52.0	96.2	88.2	95.8	88.5	95.2	79.9	83.4
Qwen3-VL-32B (Bai et al., 2025)	**93.4**	**71.3**	**92.8**	74.3	78.0	56.1	95.5	88.8	97.2	88.5	92.6	78.6	85.3
GroundGUI-8B (Ours)	92.3	69.9	89.0	**80.3**	**81.7**	61.7	**96.8**	**89.7**	**97.5**	**91.3**	94.8	**82.5**	**86.8**

Table 4: Performance comparison on **ScreenSpot-Pro** benchmark. **Bold** values = best results.

Model	CAD		Dev.		Creative		Scientific		Office		OS		Avg.
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Qwen3-VL-2B (Bai et al., 2025)	31.0	15.6	55.2	11.7	59.1	16.1	64.6	22.7	72.3	34.0	59.8	23.6	41.9
InfiGUI-3B (Liu et al., 2026a)	50.8	25.0	64.9	20.0	51.5	16.8	68.8	32.7	70.6	32.1	49.5	19.7	45.2
Ferret-UI Lite (You et al., 2024)	-	-	-	-	-	-	-	-	-	-	-	-	53.3
UI-TARS-7B (Qin et al., 2025)	20.8	9.4	58.4	12.4	50.0	9.1	63.9	31.8	63.3	20.8	30.8	16.9	35.7
Phi-Ground (Zhang et al., 2025)	26.9	17.2	70.8	16.7	56.6	13.3	58.0	29.1	76.4	44.0	55.1	25.8	43.2
GUI-Actor-7B (Wu et al., 2025a)	47.7	9.4	59.1	15.9	59.6	16.1	70.1	25.5	69.5	41.5	55.1	19.1	44.6
SE-GUI-7B (Yuan et al., 2025)	51.3	14.1	68.2	19.3	57.6	9.1	75.0	28.2	78.5	43.4	49.5	25.8	47.2
GUI-G^2-7B (Tang et al., 2026)	55.8	12.5	68.8	17.2	57.1	15.4	77.1	24.5	74.0	32.7	57.9	21.3	47.5
Qwen3-VL-8B (Bai et al., 2025)	46.7	10.9	79.2	23.4	68.2	14.0	73.6	30.0	76.3	30.2	65.4	21.3	49.9
OpenCUA-7B (Wang et al., 2025c)	-	-	-	-	-	-	-	-	-	-	-	-	50.0
GTA1-7B (Yang et al., 2026)	53.3	17.2	66.9	20.7	62.6	18.9	76.4	31.8	82.5	**50.9**	48.6	25.9	50.1
UI-Venus-7B (Gu et al., 2025)	60.4	21.9	74.7	24.1	63.1	14.7	76.4	31.8	75.7	41.5	49.5	22.5	50.8
InfiGUI-G1-7B (Liu et al., 2026a)	57.4	23.4	74.7	24.1	64.6	18.2	80.6	31.8	75.7	39.6	57.0	29.2	51.9
Holo2-8B (H Company, 2025)	-	-	-	-	-	-	-	-	-	-	-	-	58.9
MolmoPoint-8B (Clark et al., 2026)	-	-	-	-	-	-	-	-	-	-	-	-	60.2
Step-GUI-8B (Yan et al., 2025)	-	-	-	-	-	-	-	-	-	-	-	-	62.6
COPC-8B (Wang et al., 2026)	-	-	-	-	-	-	-	-	-	-	-	-	58.7
Qwen3-VL-32B (Bai et al., 2025)	60.4	28.1	69.5	22.1	**75.8**	25.2	**84.7**	25.5	85.9	43.4	62.6	15.7	54.9
OpenCUA-32B (Wang et al., 2025c)	-	-	-	-	-	-	-	-	-	-	-	-	55.3
GUI-Owl-32B (Wang et al., 2025b)	62.4	28.1	**84.4**	39.3	65.2	18.2	82.6	39.1	81.4	39.6	70.1	36.0	58.0
ScaleCUA-32B (Liu et al., 2026b)	60.4	39.1	75.3	35.2	73.2	30.8	76.4	**46.4**	81.4	49.1	63.6	41.6	59.2
UGround-V1-72B (Gou et al., 2025)	16.8	4.7	55.8	4.8	54.0	10.5	70.8	22.7	61.0	18.9	40.2	7.9	34.5
UI-TARS-72B (Qin et al., 2025)	18.8	12.5	63.0	17.2	57.0	15.4	64.6	20.9	63.3	26.4	42.1	15.7	38.1
UI-Venus-72B (Gu et al., 2025)	-	-	-	-	-	-	-	-	-	-	-	-	61.9
GroundGUI-8B	**67.5**	**40.6**	**84.4**	**42.1**	74.2	**35.0**	80.6	41.8	**89.8**	**50.9**	**71.0**	**44.9**	**63.9**

instructions, emphasizing reasoning over element semantics and relative spatial relationships. Baseline scores are taken from the corresponding publications.

5.3 Comparison with Other Models

As shown in Tables 2–6, our GroundGUI-8B matches or outperforms all listed open-source 7B-scale baselines across all evaluated benchmarks, demonstrating strong GUI element grounding. Compared with the base model Qwen3-VL-8B (Bai et al., 2025), GroundGUI-8B achieves consistent performance gains: 3.5 points on ScreenSpot-V2, 5.5 on MMBench-GUI L2, 14.0 on ScreenSpot-Pro, 10.6

Table 5: Performance comparison on **OSWorld-G** benchmark. **Bold** values = best results.

Model	Text Matching	Element Recognition	Layout Understanding	Fine-grained Manipulation	Refusal	Avg
JEDI-3B (Xie et al., 2025)	67.4	53.0	53.8	44.3	7.4	50.9
Qwen3-VL-2B (Bai et al., 2025)	61.7	45.8	54.2	39.6	-	45.9
OS-ATLAS-7B (Wu et al., 2025b)	44.1	29.4	35.2	16.8	7.4	27.7
UGround-7B (Gou et al., 2025)	51.3	40.3	43.5	24.8	0.0	36.4
Aguvis-7B (Xu et al., 2024)	55.9	41.2	43.9	28.2	0.0	38.7
UI-TARS-7B (Qin et al., 2025)	60.2	51.8	54.9	35.6	0.0	47.5
UI-TARS-1.5-7B (Qin et al., 2025)	36.8	62.7	62.2	50.8	0.0	52.8
JEDI-7B (Xie et al., 2025)	65.9	55.5	57.7	46.9	7.4	54.1
Qwen3-VL-8B (Bai et al., 2025)	69.0	55.5	59.7	47.7	-	54.8
GTA1-7B (Yang et al., 2026)	42.1	65.7	62.7	56.1	0.0	55.1
GUI-Owl-7B (Wang et al., 2025b)	64.8	63.6	61.3	41.0	-	55.9
UI-Venus-7B (Gu et al., 2025)	74.6	60.5	61.5	45.5	-	58.8
Holo2-8B (H Company, 2025)	-	-	-	-	-	62.2
MolmoPoint-8B (Clark et al., 2026)	-	-	-	-	-	62.4
COPC-8B (Wang et al., 2026)	-	-	-	-	-	59.6
OpenCUA-32B (Wang et al., 2025c)	-	-	-	-	-	59.6
GUI-Owl-32B (Wang et al., 2025b)	67.0	64.5	67.2	45.6	-	58.0
Qwen3-VL-32B (Bai et al., 2025)	72.8	63.3	66.4	51.7	-	60.6
GTA1-32B (Yang et al., 2026)	63.2	**78.4**	**73.3**	**65.2**	0.0	65.2
GroundGUI-8B	**77.0**	68.5	66.4	57.2	**22.2**	**65.4**

Table 6: Performance comparison on **UI-Vision** benchmark. **Bold** values = best results.

Model	Basic	Functional	Spatial	Avg
SeeClick (Cheng et al., 2024)	9.4	4.7	2.1	5.4
Qwen3-VL-2B (Bai et al., 2025)	0.0	19.2	0.1	6.2
OS-ATLAS-7B (Wu et al., 2025b)	12.2	11.2	3.7	9.0
UGround-V1-7B (Gou et al., 2025)	15.4	17.1	6.3	12.9
Qwen3-VL-8B (Bai et al., 2025)	25.0	27.9	1.2	17.5
UI-TARS-7B (Qin et al., 2025)	20.1	24.3	8.4	17.6
InfiGUI-G1-3B (Liu et al., 2026a)	31.2	28.0	8.2	22.0
UI-TARS-1.5-7B (Qin et al., 2025)	28.8	27.5	10.7	22.3
InfiGUI-G1-7B (Liu et al., 2026a)	36.2	31.9	11.5	26.1
UI-Venus-7B (Gu et al., 2025)	36.1	32.8	11.9	26.5
Phi-Ground (Zhang et al., 2025)	36.8	37.1	7.6	27.2
UI-TARS-72B (Qin et al., 2025)	31.4	30.5	14.7	25.5
Qwen3-VL-32B (Bai et al., 2025)	32.8	34.2	14.7	26.9
UI-Venus-72B (Gu et al., 2025)	45.6	42.3	23.7	36.8
GroundGUI-8B	**46.3**	**43.5**	**26.3**	**38.3**

on OSWorld-G, and 20.8 on UI-Vision. These improvements are consistent with contributions from our ScreenRef data and D2RL, which jointly improve grounding across the evaluated benchmarks. Moreover, GroundGUI-8B surpasses Step-GUI-8B (Yan et al., 2025) by 1.3 points on ScreenSpot-Pro and 1.2 on MMBench-GUI L2. GroundGUI-8B delivers notably better results than Holo2-8B and MolmoPoint-8B on OSWorld-G, ScreenSpot-Pro, and ScreenSpot-V2. Notably, Step-GUI and Holo2 keep their training datasets non-public, whereas we plan to release our annotations, code, and redistributable data to facilitate further research. Beyond 8B counterparts, GroundGUI-8B also achieves stronger results than multiple 32B-scale models, including GUI-Owl-32B (Wang et al., 2025b), ScaleCUA-32B (Liu et al., 2026b), OpenCUA-32B (Wang et al., 2025c), and Qwen3-VL-32B. These results indicate that GroundGUI-8B attains strong reported performance with far fewer parameters than these models.

5.4 Comparison with Other Datasets

To directly compare with other open-source datasets, we train Qwen3-VL-8B on previous state-of-the-art grounding datasets, including ScaleCUA and GroundCUA. We explore the best training parameters for each dataset. As GroundCUA contains many images from UI-Vision, we omit the

Table 7: Qwen3-VL-8B fine-tuned on different grounding datasets under recommended settings for each. UI-Vision is omitted due to source overlap with GroundCUA. **Bold** values = best results.

Model	SS-Pro	OSW-G	MMB-GUI L2	SS-V2	Average
Qwen3-VL-8B (reported) (Zhao et al., 2026b)	49.9	54.8	81.3	91.7	69.4
+ ScaleCUA SFT	52.8	61.7	83.6	93.6	72.9
+ GroundCUA SFT	58.5	59.6	**84.9**	91.9	73.7
+ SCREENREF SFT (ours)	**62.4**	**64.9**	84.5	**93.9**	**76.4**

Table 8: Ablation study of Distillation-to-Reinforcement Learning on all GUI grounding benchmarks. RLOO† denotes a model first trained with SFT using 12K RL data and then optimized with RLOO. **Bold** values = best results.

Model	SS-V2	SS-Pro	MMB-GUI L2	UI-Vision	OSW-G	Average
SFT	93.9	62.4	84.5	34.3	64.9	68.0
+ OPSD	95.0	62.6	85.8	37.0	65.1	69.1
+ RLOO	95.0	63.4	85.8	36.5	64.0	68.9
+ RLOO†	95.0	62.8	86.4	37.5	63.8	69.1
+ D2RL	**95.2**	**63.9**	**86.8**	**38.3**	**65.4**	**69.9**

score on UI-Vision for fair comparison. As shown in Table 7, averaged over four benchmarks, the SFT-only model trained on ScreenRef yields a 3.5-point gain over ScaleCUA and 2.7 points over GroundCUA. The results suggest that ScreenRef is more effective than these datasets on the evaluated benchmarks. On OSWorld-G, the ScreenRef model notably outperforms those trained on ScaleCUA and GroundCUA, by 3.2 and 5.3 points, respectively. A likely contributing factor is that ScreenRef provides refusal samples, which are absent in the other datasets. On ScreenSpot-Pro, the ScreenRef SFT model presents a larger advantage over ScaleCUA and GroundCUA, likely because our dataset comprises data across a broader diversity and higher annotation accuracy.

5.5 ABLATION STUDY

We investigate the effectiveness of our method's key components (Table 8), where all post-training variants start from the same SFT checkpoint and use the same post-training data, and the single-stage variants (+OPSD and +RLOO) are each trained for 2 epochs. **OPSD:** Adding OPSD to SFT improves average score by 1.1 points, indicating that OPSD tailored for grounding improves the model via on-policy token-level supervision. **RLOO:** Applying RLOO after SFT yields a 0.9-point average gain. This result indicates that RLOO improves the SFT-only model by optimizing the policy toward target rewards. **D2RL:** D2RL surpasses SFT-only, SFT+OPSD, and SFT+RLOO on all benchmarks, supporting the method's effectiveness. Note that D2RL also outperforms RLOO† by 0.8 points. These results demonstrate that the benefit of D2RL is not due to extra exposure to the post-training data. D2RL's advantage comes from the stronger initial policy that OPSD provides.

6 CONCLUSION

We introduce ScreenRef, a large-scale, high-quality GUI grounding dataset of 586K screenshots and 633K referring expressions across desktop, web, and mobile environments. Using ScreenRef, we construct a suite of real-world instruction tasks for multimodal grounding. We also present GroundGUI-8B, trained via SFT followed by D2RL, which sequentially applies OPSD and RLOO. Across 5 benchmarks, GroundGUI substantially outperforms models trained on existing open-source datasets, while achieving performance competitive with or superior to leading models trained on closed-source data. Our findings suggest that data quality and coverage—rather than sheer volume alone—are important contributors to robust GUI grounding. We will release checkpoints, code, annotations, and redistributable data, subject to source-platform terms and privacy review, aiming to advance grounding as a foundational capability for generalist end-to-end GUI agents.

AI USE STATEMENT

In this work, we used generative AI models to generate synthetic datasets and to clean and reformat dataset content, as described in Section 3: Qwen3-VL-235B-A22B-Instruct generated grounding instructions from denoised metadata; Qwen3.8-27B verified instruction–target consistency; Qwen3.6-27B filtered non-tutorial video metadata; GPT-5.5 produced task segmentation, action annotations, and referring expressions from video frame pairs and transcripts; and Qwen3.7-Plus and Claude Opus 5 produced target coordinates with confidence gating. OmniParser V2 and YOLOv8 were used as non-generative detectors. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, write proofs, propose or refine hypotheses, design or evaluate the research methodology, translate, perform qualitative analysis, or interpret results. Additionally, we used generative AI tools to aid and polish the writing of this manuscript. All AI-generated data passed the automated verification stages described in Section 3, and all AI-assisted text was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

ScreenRef is built from screenshots of live GUI environments and from frames of publicly available tutorial videos; no human subjects were recruited. Screenshots of real interfaces may incidentally contain personal or account information, and we have not yet performed a systematic human privacy audit; a privacy review will precede any data release, and non-redistributable material will be excluded. We plan to release annotations, processing code, and all redistributable data, subject to source-platform terms, licenses, and privacy review. Improved GUI grounding can enable automated interaction with software, including unapproved automation on sensitive services; the released artifacts are intended for research on grounding, and their documentation will state this intended use.

REPRODUCIBILITY STATEMENT

The dataset construction pipeline is described in Section 3 and Figure 1, including the models, thresholds (IoU 0.5, confidence $\tau = 0.95$, at most 60 candidates per video), and verification stages used; D2RL is specified in Section 4 with Equations 1–3 and Figure 2; training data sizes, hardware, epochs, batch size, learning rate, and rollout count are given in Section 5.1; benchmarks and the treatment of baseline scores are described in Section 5.2; dataset statistics and examples are in Appendices A–D. The prompts, filtering rules, per-stage filtering statistics, and the full training configuration, including the position-aware weighting strength β, will be released together with the code.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, et al. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731*, 2021.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Guozhi Wang, Dingyu Zhang, Shuai Ren, and Hongsheng Li. Amex: Android multi-annotation expo dataset for mobile gui agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2138–2156, 2025.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the*

62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 9313–9332, 2024.

Christopher Clark, Yue Yang, Jae Sung Park, Zixian Ma, Jieyu Zhang, Rohun Tripathi, Mohammadreza Salehi, Sangho Lee, Taira Anderson, Winson Han, et al. Molmopoint: Better pointing for vlms with grounding tokens. *arXiv preprint arXiv:2603.28069*, 2026.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854, 2017.

Aarash Feizi, Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Kaixin Li, Rabiul Awal, Xing Han Lù, Johan S Obando Ceron, Juan A Rodriguez, Nicolas Chapados, et al. Grounding computer use agents on human demonstrations. In *International Conference on Learning Representations*, volume 2026, pp. 5181–5215, 2026.

Changlong Gao, Zhangxuan Gu, Yulin Liu, Xinyu Qiu, Shuheng Shen, Yue Wen, Tianyu Xia, Zhenyu Xu, Zhengwen Zeng, et al. Ui-venus-1.5 technical report. *arXiv preprint arXiv:2602.09082*, 2026.

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

Zhangxuan Gu, Zhengwen Zeng, Zhenyu Xu, Xingran Zhou, Shuheng Shen, Yunfei Liu, Beitong Zhou, Changhua Meng, Tianyu Xia, Weizhi Chen, Yue Wen, Jingya Dou, Fei Tang, Jinzhen Lin, Yulin Liu, Zhenlin Guo, Yichen Gong, Heng Jia, Changlong Gao, Yuan Guo, Yong Deng, Zhenyu Guo, Liang Chen, and Weiqiang Wang. Ui-venus technical report: Building high-performance ui agents with rft. *arXiv preprint arXiv:2508.10833*, 2025.

H Company. Holo2 - open foundation models for navigation and computer use agents, 2025. URL https://huggingface.co/collections/Hcompany/holo2.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290. IEEE, 2024.

Junlong Li, Chao Hao, Lap-Pui Chau, and Yi Wang. GUI-C^2: Coarse-to-fine gui grounding via difficulty-aware reinforcement learning. *arXiv preprint arXiv:2605.30884*, 2026.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025. doi: 10.1145/3746027.3755688.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8198–8210, 2020a. doi: 10.18653/v1/2020.acl-main.729.

Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5495–5510, 2020b. doi: 10.18653/v1/2020.emnlp-main.443.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19498–19508. IEEE, 2025.

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xueyu Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, et al. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32267–32275, 2026a.

Zhaoyang Liu, JingJing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, Xuehui Wang, Qiushi Sun, Shi Liu, Weiyun Wang, et al. Scalecua: Scaling open-source computer use agents with cross-platform data. In *International Conference on Learning Representations*, volume 2026, pp. 62707–62766, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Pengxiang Zhao, Guangyi Liu, et al. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17608–17616, 2026.

Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025.

Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A. Rodriguez, Montek Kalsi, Rabiul Awal, Nicolas Chapados, M. Tamer Ozsu, Aishwarya Agrawal, David Vazquez, Christopher Pal, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. *arXiv preprint arXiv:2503.15661*, 2025.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026a. URL `https://qwen.ai/blog?id=qwen3.6-27b`.

Qwen Team. Qwen3.7-Plus: Multimodal agent intelligence, May 2026b. URL `https://qwen.ai/blog?id=qwen3.7-plus`.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL `https://qwen.ai/blog?id=qwen3.8`.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728, 2023.

Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. arxiv 2023. *arXiv preprint arXiv:2305.09972*, 10, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, et al. 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)*, pp. 5555–5579, 2025.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, et al. GUI-G^2: Gaussian reward modeling for gui grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33214–33222, 2026.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025a.

Junyang Wang, Lu Chen, Wentao Zhang, Yifan Ye, Xinyu Wang, Yuxuan Song, Chang Fan, Zilin Zheng, Xi Guo, Jun Lu, Xiang Li, Pengwei Xie, Shumin Deng, Ningyu Zhang, and Huajun Chen. Mobile-Agent-v3: Fundamental agents for GUI automation. *arXiv preprint arXiv:2502.17066*, 2025b.

Wenkai Wang, Xiyun Li, Hongcan Guo, Wenhao Yu, Tianqing Fang, Haitao Mi, Dong Yu, and Shengyu Zhang. Measure twice, click once: Co-evolving proposer and visual critic via reinforcement learning for gui grounding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20964–20984, 2026.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Wu, et al. Opencua: Open foundations for computer-use agents. *Advances in Neural Information Processing Systems*, 38:139756–139806, 2025c.

Xuehui Wang, Zhenyu Wu, JingJing Xie, Zichen Ding, Bowen Yang, Zehao Li, Zhaoyang Liu, Qingyun Li, Xuan Dong, Zhe Chen, et al. Mmbench-gui: Hierarchical multi-platform evaluation framework for gui agents. *arXiv preprint arXiv:2507.19478*, 2025d.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2023.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. Gui-actor: Coordinate-free visual grounding for gui agents. *Advances in Neural Information Processing Systems*, 38:15101–15128, 2025a.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: Foundation action model for generalist gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 5090–5108, 2025b.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. Spotlight.

Weimin Xiong, Shuhao Gu, Bowen Ye, Zihao Yue, Lei Li, Feifan Song, Sujian Li, and Hao Tian. Video2gui: Synthesizing large-scale interaction trajectories for generalized gui agent pretraining. *arXiv preprint arXiv:2605.14747*, 2026.

Haiyang Xu, Xi Zhang, Haowei Liu, Junyang Wang, Zhaozai Zhu, Shengjie Zhou, Xuhao Hu, Feiyu Gao, Junjie Cao, Zihua Wang, et al. Mobile-agent-v3. 5: Multi-platform fundamental gui agents. *arXiv preprint arXiv:2602.16855*, 2026.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*, 2024.

Haolong Yan, Jia Wang, Xin Huang, Yeqing Shen, Ziyang Meng, Zhimin Fan, Kaijun Tan, Jin Gao, Lieyu Shi, Mi Yang, et al. Step-gui technical report. *arXiv preprint arXiv:2512.15431*, 2025.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Caiming Xiong, and Junnan Li. Gta1: Gui test-time scaling agent. In *International Conference on Learning Representations*, 2026.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22418–22433, 2025. doi: 10.18653/v1/2025.findings-acl.1152.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, pp. 240–255. Springer, 2024.

Wenwen Yu, Zhibo Yang, Jianqiang Wan, Sibo Song, Jun Tang, Wenqing Cheng, Yuliang Liu, and Xiang Bai. Omniparser v2: Structured-points-of-thought for unified visual text parsing and its generality to multimodal large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. Se-gui: Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*, 2025.

Jiwen Zhang, Jihao Wu, Teng Yihua, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. Android in the zoo: Chain-of-action-thought for gui agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12016–12031, 2024.

Miaosen Zhang, Ziqiang Xu, Jialiang Zhu, Qi Dai, Kai Qiu, Yifan Yang, Chong Luo, Tianyi Chen, Justin Wagle, Tim Franklin, and Baining Guo. Phi-ground tech report: Advancing perception in gui grounding. *arXiv preprint arXiv:2507.23779*, 2025.

Yan Zhang, Daiqing Wu, Huawen Shen, Can Ma, and Yu Zhou. Learn where to click from yourself: On-policy self-distillation for gui grounding. *arXiv preprint arXiv:2605.00642*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026a.

Zhongyin Zhao, Yuan Liu, Yikun Liu, Haicheng Wang, Le Tian, Xiao Zhou, Yangxiu You, Zilin Yu, Yang Yu, and Jie Zhou. Points-gui-g: Gui-grounding journey. *arXiv preprint arXiv:2602.06391*, 2026b.

Hanzhang Zhou, Xu Zhang, Panrong Tong, et al. MAI-UI technical report: Real-world centric foundation GUI agents. *arXiv preprint arXiv:2502.16275*, 2025.

APPENDIX

A ADDITIONAL SCREENREF EXAMPLES

We provide additional qualitative examples of the four ScreenRef task types across desktop, web, and mobile environments. Each answer is a tool call with an action and an integer coordinate pair normalized to $[0, 1000]$ along the image width and height; refusal samples use an abstain action (Figures 3–5).

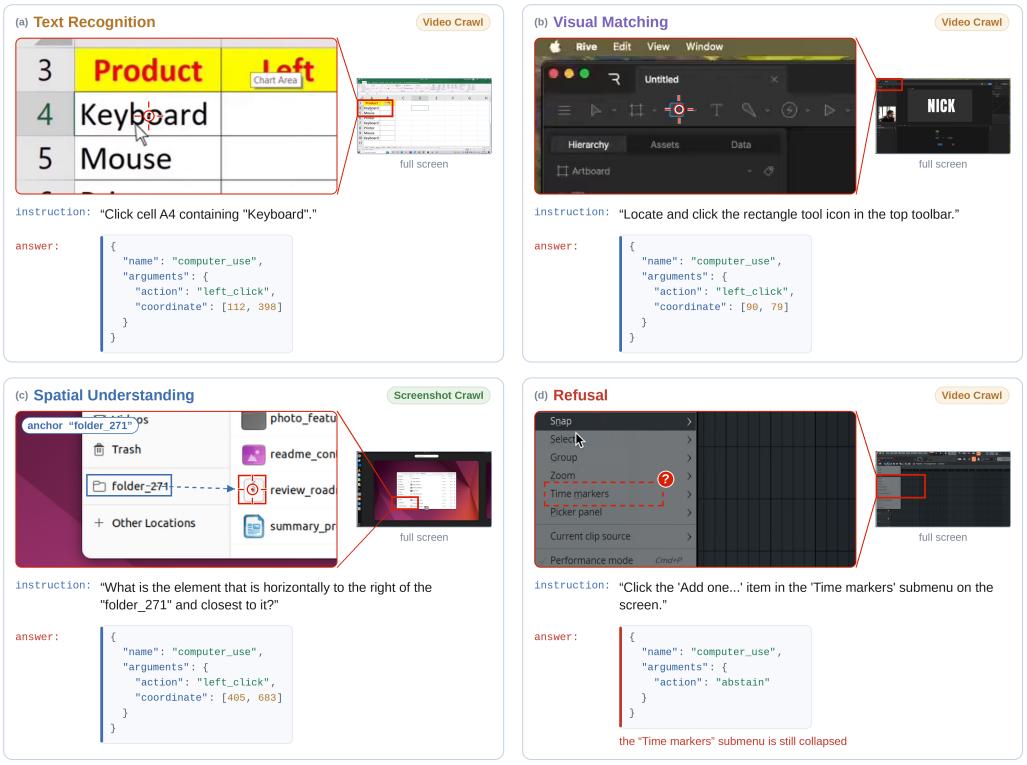

Figure 3: Examples of the four ScreenRef task types in desktop environments.

B ADDITIONAL DETAILS FOR DATASET COMPARISON

Figures for prior datasets are reported as published rather than recomputed here. For Images and Res Range we align with the comparison in Table 1 of Feizi et al. (2026), the paper that introduces GroundCUA and is itself a row here, so that these columns rely on consistent measurements across datasets rather than totals produced under different protocols; where that comparison does not cover a dataset we use the figure reported in its own paper. They differ in places from a dataset's own stated total, which may reflect different filtering. Instr. is each paper's own reported count, left blank where none is given rather than derived from element counts. Platform follows each paper's account of its data, and where a reported figure covers only part of a dataset the marks follow that part.

Refusal marks grounding samples whose referred target is absent from the screenshot; a task-level infeasible action, which several agent datasets provide, is not counted. A pipeline that derives every instruction from an element present in the screenshot yields no such sample, which is the case for every prior dataset here except JEDI, which pairs instructions with unrelated screenshots. Figures for SCREENREF are measured on the final dataset.

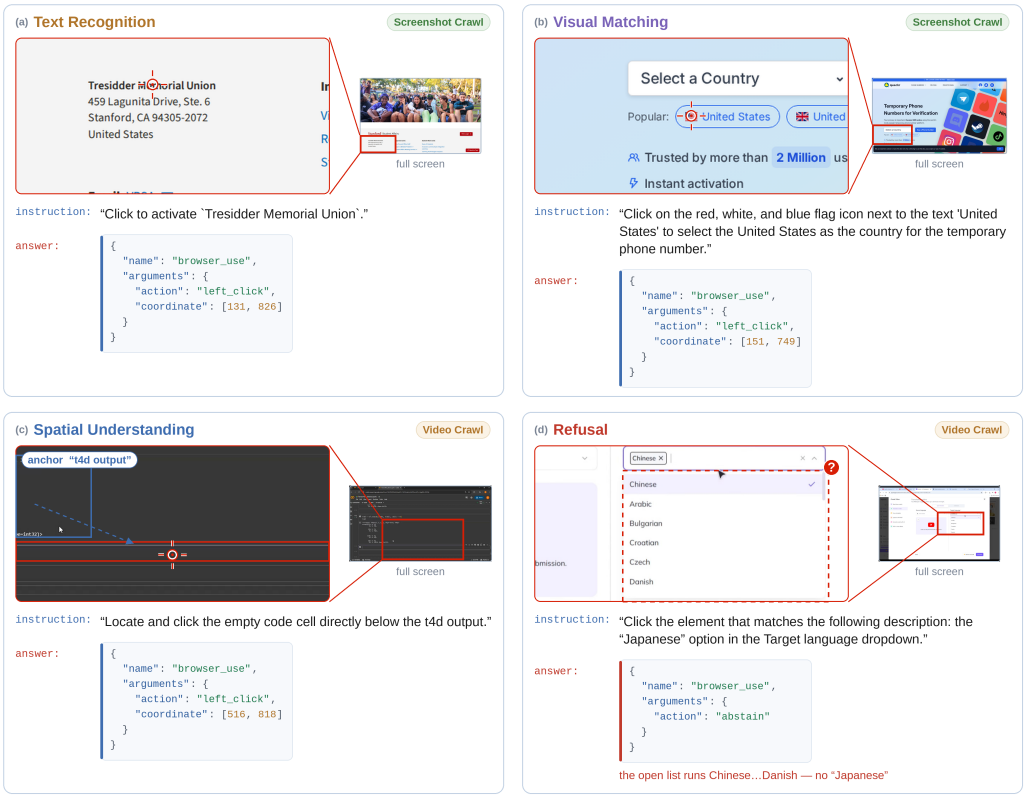

Figure 4: Examples of the four ScreenRef task types in web environments.

Table 9: Distribution of grounding capabilities across platforms.

Platform	Text	Visual	Spatial	Refusal	Total
Desktop	128,018	61,951	63,291	13,912	267,172
Web	167,115	62,294	67,799	6,582	303,790
Mobile	24,160	12,526	23,740	1,981	62,407
Total	**319,293**	**136,771**	**154,830**	**22,475**	**633,369**

C Additional ScreenRef Statistics

We provide additional statistics on the distribution of grounding capabilities in ScreenRef. Table 9 summarizes the number of grounding instances across desktop, web, and mobile platforms for the four capability categories: Text Recognition, Visual Matching, Spatial Understanding, and Refusal. Text Recognition constitutes the largest portion of ScreenRef, accounting for 50.4% of all grounding instances, followed by Spatial Understanding (24.4%), Visual Matching (21.6%), and Refusal (3.5%). The capability distribution also varies across platforms. Text Recognition represents the largest category on both web (55.0%) and desktop (47.9%) interfaces, whereas the mobile subset is more balanced between Text Recognition (38.7%) and Spatial Understanding (38.0%). Desktop interfaces contain the highest proportion of Visual Matching instances (23.2%) and Refusal instances (5.2%), reflecting the greater prevalence of visually identified controls and absent-target scenarios in desktop environments.

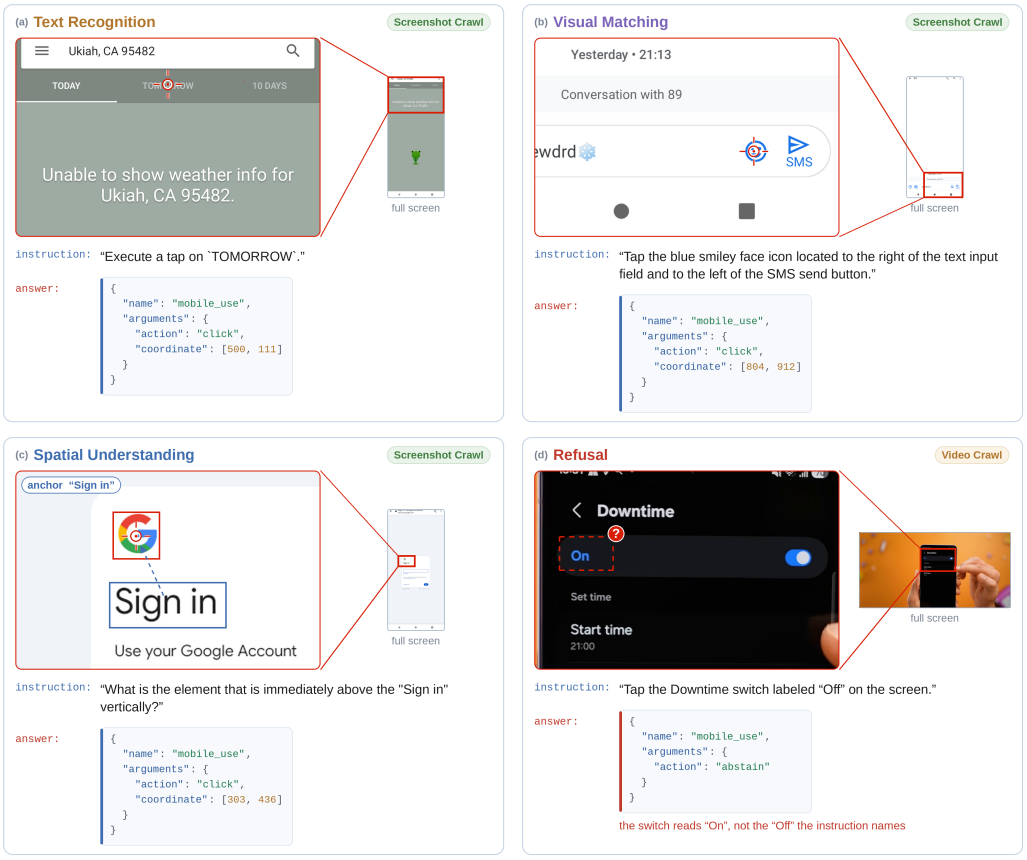

Figure 5: Examples of the four ScreenRef task types in mobile environments.

D ELEMENT METADATA FROM LIVE SCREENSHOT CRAWLING

Part of ScreenRef is built from live screenshot crawling, where we capture screenshots from real web pages, desktop applications, and Android applications. Alongside each screenshot, we record element metadata from the platform's native interface: the DOM on web pages, the accessibility tree on desktop, and the XML layout on Android. Each element record contains a bounding box in the pixel coordinates of the screenshot, a text string, and a functional category such as Button, Input Elements, or Visual Elements. The text is the element's visible text or, if the element displays none, its accessibility name; the category may be absent for some Android elements. Every record contains non-empty text, so the metadata covers the elements that the interface exposes with a text or an accessibility name, rather than every visual object on the screen. Before the metadata is used to generate grounding instructions, it is filtered to remove or refine unreliable screenshots and boxes. Figure 6 shows real examples of the filtered metadata.

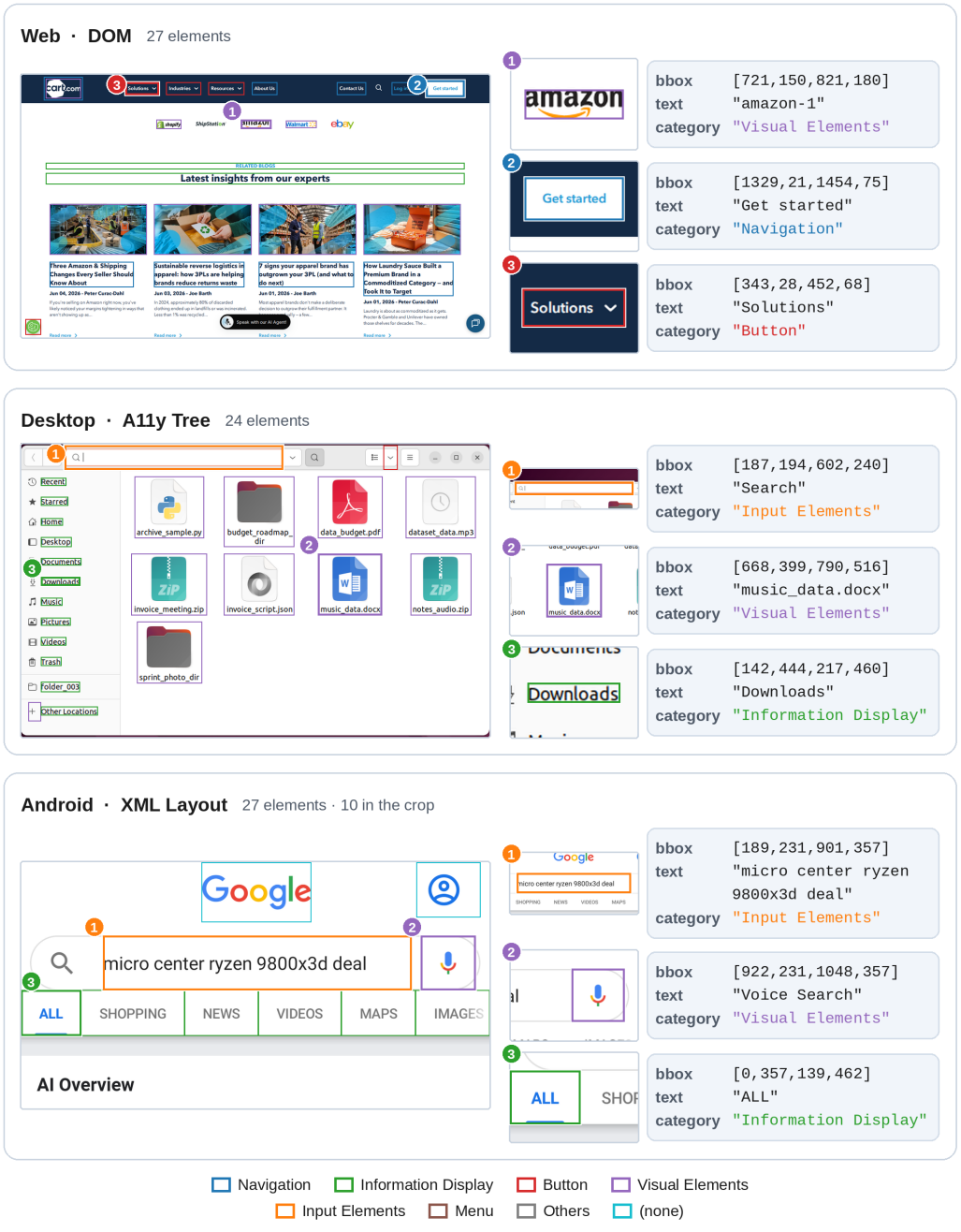

Figure 6: Examples of element metadata from live screenshot crawling. Left: a real screenshot, cropped to the region containing metadata, with all element boxes in this region that survive filtering, colored by category (“(none)” marks elements without a category). Right: three numbered elements from the same screenshot, zoomed in, with their stored records (bounding box, text, and category).