WinDOM: Self-Family Distillation for Small-Model GUI Grounding

Chengheng Li Chen [1 2] **Zhiqian Zhou** [1] **Hao Chen** [3] **Nicolas Chauvin** [2]

Abstract

Small (∼2B) GUI-grounding agents are attractive for on-device deployment, accessibility tooling, and low-cost iteration, but at this scale they face two open recipe questions: how to obtain bounding-box training data without expensive human annotation, and how to combine supervised fine-tuning with reinforcement learning. We address both, with the explicit goal of pushing small-model performance rather than scaling up. **WinDOM** is a 54,425-record grounding corpus harvested by driving an open-source Windows 11 web reimplementation under headless Playwright, with bounding boxes read directly off the DOM and no OCR or human annotation. *Self-Family Distillation* (SFD) is a single rejection-sampling cold-start parameterised only by the teacher choice: either an EMA of the student (no external model) or a frozen larger same-family teacher. We then treat the saturation depth of the SFD cold-start as an explicit GRPO hyperparameter. On a Qwen3.5-2B student, the under-saturated cold-start is a better GRPO initialiser than the converged one: SFD-4B with Early-init RL gains +5.4 OOD-mean (+3.5 ScreenSpot-Pro, +7.0 OSWorld-G, +5.8 ScreenSpot-V2) over the base. The same-size EMA mode lands within roughly one OOD-mean point of the cross-size 4B variant (65.2 vs 66.3) without an external teacher.

1. Introduction

A GUI agent that cannot reliably click the right pixel cannot do anything else: at small (∼2B) scale, grounding accuracy is both the capability bottleneck and the surface on which every downstream tool-call or action-guard ultimately fires. Small-model grounding runs into two costs. **Data:** professional corpora such as ScreenSpot-Pro (Li et al., 2025) need expensive human bounding-box annotation, and OCR-mined alternatives (Cheng et al., 2024) inject label noise. **Post-training:** there is no consensus on how to combine SFT with RL, with recipes split between skipping SFT (Shao et al., 2024; Yang et al., 2026) and launching GRPO from a fully converged SFT base (Liu et al., 2025a); parallel evidence (Liu et al., 2025b; Jin et al., 2025) points to the *saturation depth* of the SFT initialisation, not its presence, as the real lever. Reliable grounding is also a safety prerequisite for agents in the wild: a model that cannot locate the intended control cannot be constrained to it, so grounding errors surface downstream as clicks on unintended, possibly harmful, UI elements rather than as clean refusals.

We address both in one pipeline. For data, we render an open-source Windows 11 web clone (Win11React) headlessly and read layout-rectangle bounding boxes straight from the DOM, eliminating OCR and human labels; the corpus, **WinDOM**, has 500 trajectories, 2,013 screenshots, and 54,425 grounded records. For post-training, we unify a frozen larger same-family teacher (cross-size) and an EMA self-teacher (same-size) (Shenfeld et al., 2026) under one click-in-bounding-box rejection-sampling cold-start we call *Self-Family Distillation* (SFD), and treat its *saturation depth* as the central GRPO hyperparameter (Figure 1: SFD-4B, then GRPO from an under-saturated step-100 checkpoint). Relevant to deploying agents in the wild, every supervisory signal here, the SFD filter and the GRPO reward, is a single deterministic point-in-box check rather than a learned reward model or LLM judge, so the whole training channel is cheap to audit and reproduce.

Contributions.

1. **WinDOM.** A 54k-record GUI grounding corpus with bounding boxes read directly from the DOM layout rectangle (no OCR, no human annotation), harvested from an open-source Windows 11 clone and released with a deterministic regeneration script.
2. **Self-Family Distillation.** One rejection-sampled cold-start unifying an EMA self-teacher (Shenfeld et al., 2026) and a frozen larger same-family teacher, set by the teacher choice alone; the teacher-free EMA variant comes within one OOD-mean point of the 4B teacher.

[1]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland [2]Logitech [3]Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Correspondence to: Chengheng Li Chen <chengheng.lichen@epfl.ch>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

3. **Cold-start depth as a GRPO hyperparameter.** Early-init RL on the SFD-4B cold-start trades a few in-distribution points for $+3.5$ ScreenSpot-Pro, $+7.0$ OSWorld-G, and $+5.8$ ScreenSpot-V2 over the Qwen3.5-2B base; Late-init RL holds in-distribution accuracy but barely moves OOD in every regime.

2. Related Work

We situate WinDOM and Self-Family Distillation against three lines of prior work: GUI-grounding corpora and small-model grounding agents, filtered teacher-student distillation in language modelling, and reinforcement learning from cold-started supervised checkpoints.

GUI grounding datasets and models. The standard evaluation suites for click-level grounding are ScreenSpot-Pro (Li et al., 2025), OSWorld-G (Xie et al., 2025), and ScreenSpot-V2 (Wu et al., 2025); on the training-corpus side, OS-Atlas (Wu et al., 2025) and Jedi (Xie et al., 2025) reach desktop scale by mining or decomposing real screenshots. WinDOM is desktop-OS-specific, reads layout-rectangle bounding boxes directly from the DOM of a deterministic Win11React renderer, and is seedable end-to-end so the corpus can be regenerated from a single seed. Our backbone is Qwen3.5-2B (Qwen Team, 2026), whose tool-calling interface emits coordinates inside a structured JSON action and removes the need for an auxiliary grounding head.

Reinforcement learning and cold-start depth. Recent GUI-RL recipes operate at $\geq$ 3B and either skip the cold-start (Yang et al., 2026) or initialise from a single fully converged supervised checkpoint (Lu et al., 2026; Zhou et al., 2025; Wang et al., 2025), treating that choice as fixed. Parallel evidence in the reasoning literature (Chu et al., 2025; Jin et al., 2025) shows that out-of-distribution accuracy peaks early in supervised training and that subsequent reinforcement learning can recover part of what supervised training has lost. We extend that observation to GUI grounding at 2B and turn *cold-start depth* into an explicitly ablated hyperparameter, mapping its effect on the in-distribution / out-of-distribution Pareto frontier.

Self-distillation and filtered training signals. SFD draws on two established lines of prior work. The same-size mode inherits from EMA self-teaching, introduced for representation learning by Mean Teacher (Tarvainen & Valpola, 2017) and BYOL (Grill et al., 2020) and for same-architecture distillation by Born-Again Networks (Furlanello et al., 2018); SDPO (Hübotter et al., 2026), Metis-SPECS (Chen et al., 2026), and the continual-learning self-distillation analysis of Shenfeld et al. (2026) carry the same idea into RL and lifelong-learning pipelines. The cross-size mode is a filtered, off-policy variant of the same-family-teacher distillation popularised for reasoning by ReST^{EM} (Singh et al., 2024), RAFT, and STaR (Zelikman et al., 2022), with the learned or LLM-judged verifier of those methods replaced by a deterministic geometric click-in-bounding-box check. Our contribution is methodological rather than algorithmic: both modes share a single rejection-sampling training script parameterised only by the teacher choice, and the resulting checkpoint is taken under-saturated by design to serve as a cold-start for downstream GRPO rather than as a terminal stage.

3. Problem Definition

We formalise grounding as click prediction on (screenshot, instruction, bounding-box) triples, and fix the policy class and accuracy metric used throughout the paper.

3.1. The grounding task

A grounding instance is a triple

$$(\mathbf{x}_v,\, \mathbf{x}_t,\, \mathcal{B}_{\text{GT}}),\quad \mathbf{x}_v \in \mathbb{R}^{H\times W\times 3},\quad \mathcal{B}_{\text{GT}} \subset [0,W]\times[0,H], \tag{1}$$

in which $\mathbf{x}_v$ is a screenshot, $\mathbf{x}_t$ is a free-form natural-language instruction, and $\mathcal{B}_{\text{GT}}$ is the axis-aligned bounding box of the screen element that $\mathbf{x}_t$ describes. The task is to map the prompt $\mathbf{x} = (\mathbf{x}_v, \mathbf{x}_t)$ to a click

$$\mathbf{p} \in [0,W]\times[0,H],\quad \mathbf{p} \in \mathcal{B}_{\text{GT}}. \tag{2}$$

We do not assume that the instruction uniquely identifies a single visual element; when an instance admits multiple plausible targets, the model is supervised and judged only against the one box that ships with the instance.

3.2. Model class

A grounding model is a vision-language policy

$$\pi_\theta(\mathbf{o} \mid \mathbf{x}),\qquad \mathbf{o} = (o_1,\ldots,o_T) \in \mathcal{V}^*, \tag{3}$$

paired with a fixed parser

$$\text{decode} : \mathcal{V}^* \to \big([0,W]\times[0,H]\big) \cup \{\perp\} \tag{4}$$

that extracts a click candidate from $\mathbf{o}$ and returns $\perp$ when the output fails to parse. The parser is part of the environment rather than the model: a non-parsable rollout counts as an incorrect click, not as a separate failure mode. In our concrete instantiation π_θ is a tool-using vision-language transformer that emits integer coordinates on a $[0,1000]^2$ scale, rescaled by (W,H) before scoring; Section 6 reports the exact emission and parsing protocol.

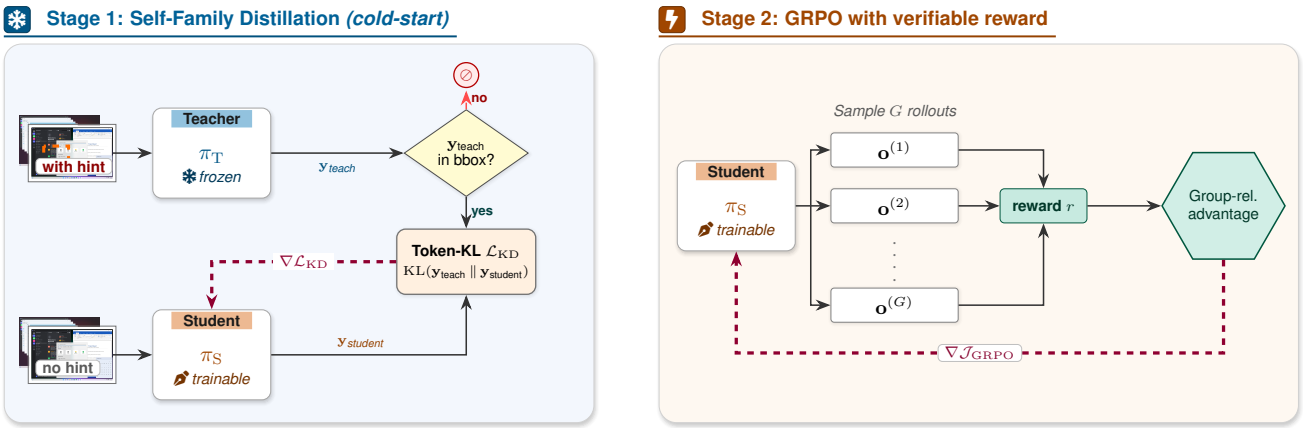

Figure 1. Pipeline of our best configuration. **Stage 1 (blue):** a frozen same-family teacher π_T receives the screenshot plus a jittered bounding-box hint; the student π_S receives only the screenshot. Teacher rollouts that miss the GT box are rejected, and the surviving ones supervise π_S via the forward-KL distillation loss $\mathcal{L}_{KD}$ (Equation (9)). **Stage 2 (orange):** initialised from the step-100 SFD checkpoint, π_S samples G=8 rollouts per prompt; group-relative advantages on a Gaussian click-in-bounding-box reward drive the GRPO update.

3.3. Click-in-target accuracy

The *click-in-target* indicator on an output $\mathbf{o}$ is

$$\mathrm{CIT}(\mathbf{o}, \mathcal{B}_{\mathrm{GT}}) \;:=\; \mathbf{1}[\mathrm{decode}(\mathbf{o}) \in \mathcal{B}_{\mathrm{GT}}]\,, \tag{5}$$

and click-in-target accuracy on a benchmark $\mathcal{D}$ is its empirical mean under greedy decoding,

$$\begin{aligned} \mathrm{Acc}(\mathcal{D}) &= \mathbb{E}_{(\mathbf{x},\mathcal{B}_{\mathrm{GT}})\sim\mathcal{D}}\big[\mathrm{CIT}(\hat{\mathbf{o}}, \mathcal{B}_{\mathrm{GT}})\big], \\ \hat{\mathbf{o}} &= \arg\max_{\mathbf{o}} \pi_\theta(\mathbf{o} \mid \mathbf{x}). \end{aligned} \tag{6}$$

This is the only metric reported in this paper. It is strictly stricter than center-distance and strictly looser than full bounding-box localisation: a click is correct iff it falls inside the rectangle, regardless of its distance to the centre, and the model is never required to recover the rectangle itself.

4. Data Collection: WinDOM

We construct WinDOM by treating an open-source browser reimplementation of Windows 11 as a self-annotating screenshot generator, recording paired (screenshot, DOM) snapshots from which bounding boxes are read directly off the document layout.

Why a browser? Operating systems are difficult to introspect: screenshots require a virtual machine, labels come from OCR or human annotation, and visual elements have no clean correspondence with code-level identifiers. A browser-based reimplementation reverses these properties: every visible element is a DOM node whose bounding box is exposed to JavaScript, whose accessibility role is stored in the document, and whose developer-supplied attributes already name the underlying action. The browser is, in effect, a self-annotating screenshot generator.

Pipeline. We instantiate the principle through Win11React, a community-maintained browser reimplementation of Windows 11 driven headlessly under Playwright. The methodology has five stages whose names match the panel titles of Figure 2; Section F lists the concrete thresholds and parameter values.

(i) Capture screen. A scene is a short script of GUI primitives authored against CSS selectors, rendered under a randomised viewport, wallpaper, and one of six layout templates per scene so the same applications appear under varied visual conditions.

(ii) Capture DOM. At every step we record a paired (screenshot, DOM) snapshot.

(iii) Bounding-box extraction. A single screenshot yields several grounding tasks: one record from the locator that produced the latest user action, plus one record per visible element whose ARIA role belongs to a fixed set of interactable roles. Each box is read directly from the document layout, so the corpus needs neither OCR nor human annotation; the layout rectangle equals the visually-clickable region only modulo CSS shadows, focus outlines, and transformed elements (see Section 7.2).

(iv) Labelling. For each extracted bounding box we prompt a small vision-language model (Gemini 2.5 Flash) with the cropped screenshot together with the element's ARIA role, accessible name, and action label, and ask it to emit a short imperative grounding instruction. Conditioning the labeller on both the image and the structured DOM signals reduces the role-name ambiguity that pure-text labelling pipelines suffer from.

(v) QA and verification. A second vision-language pass

receives the screenshot, the candidate instruction, and the bounding box, and classifies the candidate as *keep*, *edit*, or *reject*; a subset of *edit* candidates is refined by an automated vision agent and a small fraction is reviewed by a human. Geometric cleaning runs alongside the QA pass to drop degenerate boxes and near-duplicates, and every retained record carries provenance flags marking which stages it has passed.

Output and splits. The release contains 500 trajectories, 2,013 screenshots, and 54,425 grounded records, partitioned at the trajectory level into training, validation, and test splits. Training in Section 6 consumes the training partition; further details on the partitioning and the optional cleaned release are deferred to Section F.

5. Training Methodology

The training pipeline has three components: a vision-language backbone (Section 5.1), three supervised cold-start regimes (Sections 5.2 and 5.3), and a GRPO stage initialised from one of the resulting checkpoints (Section 5.4). We call a cold-start *saturated* when its in-distribution training-accuracy curve has plateaued and *early* when taken at a pre-plateau step; *Late-init RL* and *Early-init RL* denote GRPO runs that share every reinforcement-learning hyperparameter and differ only in that choice.

5.1. Backbone assumptions

The methodology assumes a vision-language policy $\pi_\theta(\mathbf{o} \mid \mathbf{x})$ in the sense of Section 3, with the additional assumption that its output sequence $\mathbf{o}$ contains a structured action sub-sequence $\mathbf{y} \subseteq \mathbf{o}$ whose tokens encode the predicted click coordinate, parsable by a fixed regular expression into a point in the same coordinate frame as $\mathcal{B}_{\text{GT}}$. We write $\mathbf{y}_{\text{teach}}$ and $\mathbf{y}_{\text{student}}$ for the action sub-sequences emitted by the teacher and student in Section 5.3, both lying inside their respective $\mathbf{o}$. The backbone is otherwise treated as a black box: no architectural modification, no vocabulary expansion, no auxiliary heads. Specific choices of backbone, parameter count, and tool-call schema are deferred to Section 6.

5.2. Cold-start regimes

We use three cold-start regimes, each implemented in the same training loop: Direct SFT (this subsection), and two modes of Self-Family Distillation (Section 5.3). Direct SFT serves as the baseline foil and is the most common cold-start in the GUI grounding literature (Cheng et al., 2024; Hong et al., 2024): given a training instance $(\mathbf{x}_v, \mathbf{x}_t, \mathcal{B}_{\text{GT}})$, the target sequence $\mathbf{y}$ is the tokenized action that emits the pixel-space center of $\mathcal{B}_{\text{GT}}$, and the loss is the standard token-level cross-entropy

$$\mathcal{L}_{\text{SFT}}(\theta) \;=\; -\,\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \sum_{t=1}^{|\mathbf{y}|} \log \pi_\theta(y_t \mid \mathbf{x}, \mathbf{y}_{<t}), \qquad (7)$$

where $\mathcal{D}$ is the WinDOM training distribution. Because the surrounding tokens are a fixed format wrapper, most of the loss mass is concentrated on the two coordinate digits.

Training is checkpointed at uniform intervals so that the cold-start can later be selected at an early or saturated point. The training-set click-in-target accuracy is a unimodal saturating curve in the training step; we treat the plateau as the saturated checkpoint and any pre-plateau point as an early checkpoint.

5.3. Self-Family Distillation

Beyond Direct SFT, our second cold-start regime is a single algorithm that we call *Self-Family Distillation* (SFD). It is filtered, off-policy distillation onto the student, parameterized by which model serves as the teacher. SFD has two natural modes, *same-size* (the teacher is an EMA of the student; no external model needed) and *cross-size* (the teacher is a frozen, stronger same-family model that shares the student's tokenizer); both use the same loss and training script.

Same-size mode (no external teacher). The teacher is an exponential moving average (EMA) of the student itself,

$$\theta_{\text{EMA}} \;\leftarrow\; \alpha\,\theta_{\text{EMA}} + (1-\alpha)\,\theta, \quad \alpha \in (0,1), \qquad (8)$$

which receives no gradient.

Cross-size mode (larger teacher). The teacher is a stronger frozen model from the same architectural family as the student. The two checkpoints are required to share the tokenizer and the special-token table so that the teacher's logits over the vocabulary can be compared to the student's without alignment. The specific student/teacher pair is deferred to Section 6.

Teacher input. In both modes the teacher receives the screenshot plus a textual *hint* consisting of the ground-truth bounding box jittered by independent integer noise of ± 30 on the $[0, 1000]^2$ scale, clamped to range; the hint is added only to the teacher's user prompt and the student is never shown it. The hint is shared by the same-size and cross-size modes so that the rejection-sampling filter (below) acts on the same teacher conditioning regardless of teacher choice; the exact prompt format is reproduced in Section H.

Filtered teacher rollouts. In both modes the teacher first generates a candidate completion $\mathbf{y}_{\text{teach}}$ from the hint-conditioned prompt. We parse a coordinate from $\mathbf{y}_{\text{teach}}$ and

Figure 2. **WinDOM data pipeline.** A scene script drives Win11React under headless Playwright with randomized viewport, wallpaper, and layout (*capture screen*). At every step we record the live DOM (*capture DOM*) and extract bounding boxes either from the latest user-action locator or from interactable ARIA roles (*bbox extract*). An LLM emits a short imperative instruction for each crop (*labeling*); a second pass classifies the candidate as *keep*, *edit*, or *reject* (*QA & verify*). Each kept pass yields one WinDOM record (image, instruction, bbox) with provenance flags.

accept the rollout only if its center lies inside the ground-truth bounding box; otherwise the example is skipped for the current step. The filter prevents distillation from spending student capacity on incorrect teacher rollouts; the empirical kept-rate of each mode is reported in Section 6.2.

Loss. Conditioned on a filtered teacher rollout $\mathbf{y}_{\text{teach}} = (y_1, \ldots, y_{|\mathbf{y}_{\text{teach}}|})$, we use the standard token-level distillation loss in cross-entropy form: forward KL from the teacher to the student, evaluated at the positions of $\mathbf{y}_{\text{teach}}$. Letting $\pi^{(t)}(\cdot) \equiv \pi(\cdot \mid \mathbf{x}, \mathbf{y}_{\text{teach},<t})$,

$$\mathcal{L}_{\text{SFD}}(\theta) \;=\; \mathbb{E}_{\mathcal{D}_{\text{kept}}} \sum_{t=1}^{|\mathbf{y}_{\text{teach}}|} \mathrm{KL}\Big(\pi_{\text{T}}^{(t)} \,\big\|\, \pi_{\theta}^{(t)}\Big), \tag{9}$$

which is equivalent to the cross-entropy of π_θ under π_{T} on teacher-sampled tokens up to a θ-independent entropy term. The KL is computed only over the positions of $\mathbf{y}_{\text{teach}}$. In same-size mode, $\pi_{\text{T}} = \pi_{\theta_{\text{EMA}}}$; in cross-size mode, π_{T} is a frozen larger same-family model. Both modes use the same loss and the same training script and differ only in the teacher assignment.

When each mode wins. The same-size mode requires no external model and no labels beyond the screenshot-click pairs already in $\mathcal{D}$, which makes it our recommended cold-start when serving a teacher alongside training would be expensive. The cross-size mode injects capability from a stronger model and is preferable when a same-family teacher is available cheaply.

5.4. Reinforcement learning with GRPO

We follow each cold-start with Group Relative Policy Optimization (GRPO). GRPO adapts the proximal-policy-optimization template to the verifiable-reward setting and removes the value network of standard PPO by computing advantages relative to a group of on-policy samples for the same prompt. We describe the algorithm in three steps.

Step 1: rollout. For each prompt $\mathbf{x}$ drawn from a training distribution $\mathcal{D}$, we sample a group of G completions independently from the rollout policy,

$$\{\mathbf{o}^{(i)}\}_{i=1}^{G} \;\sim\; \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x}).$$

Sampling uses a fixed temperature and a fixed maximum completion length. Each completion is scored by the verifiable reward

$$\begin{aligned} r(\mathbf{o}; \mathbf{x}, \mathcal{B}_{\text{GT}}) = 0.1 \cdot \mathbf{1}[\mathbf{o} \text{ parses}] \\ + \exp\!\left(-\tfrac{1}{2}\left[\tfrac{(\Delta x)^2}{\sigma_x^2} + \tfrac{(\Delta y)^2}{\sigma_y^2}\right]\right), \end{aligned} \tag{10}$$

where $(\Delta x, \Delta y)$ is the offset from the predicted click to the ground-truth box center and (σ_x, σ_y) are half the bounding-box width and height. All four quantities are in the absolute screen-pixel frame of the training screenshot, whose resolution (W, H) varies per record because WinDOM draws viewports from a weighted pool (Section F); the model emits integer coordinates on the $[0, 1000]^2$ scale of Section 3, which the reward function rescales by the per-record (W, H) before computing $(\Delta x, \Delta y)$, so σ_x and σ_y always live in the same pixel frame as the bounding box. The first term encourages parsable output; the second term, with peak value one at the center, is a Gaussian-shaped click-in-target reward (Zhou et al., 2025) that preserves variance in the group-relative advantage even when most rollouts land inside the bounding box.

Step 2: group-normalized advantage. GRPO replaces a learned value baseline with the empirical mean of the group's rewards. The advantage of the i-th completion is the standardized reward,

$$\hat{A}^{(i)} \;=\; \frac{r^{(i)} - \mu(\{r^{(j)}\}_{j=1}^{G})}{\sigma(\{r^{(j)}\}_{j=1}^{G}) + \epsilon}, \tag{11}$$

where μ and σ are the group mean and standard deviation and a small constant guards against zero variance. When all G completions for a prompt receive the same reward, $\sigma \to 0$ and the advantage degenerates; the Gaussian-shaped reward in Equation (10) mitigates this by preserving variance across rollouts even when most of them land inside the bounding box.

Step 3: clipped surrogate update. Let $\rho^{(i)}(\theta) = \pi_\theta(\mathbf{o}^{(i)} \mid \mathbf{x})/\pi_{\theta_{\text{old}}}(\mathbf{o}^{(i)} \mid \mathbf{x})$ denote the importance ratio. We maximize

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Big[\frac{1}{G}\sum_{i=1}^{G}\min\big(\rho^{(i)}\hat{A}^{(i)}, \text{clip}(\rho^{(i)}, 1-\varepsilon, 1+\varepsilon)\,\hat{A}^{(i)}\big)\Big] \tag{12}$$

with a fixed clip range $\varepsilon = 0.2$ (Section C). We do not add a Kullback-Leibler regularizer toward the cold-start policy ($\beta = 0$); the cold-start enters only as the initialization of θ.

The choice of group size and sampling temperature trades off rollout cost against the variance of the empirical baseline in Equation (11); concrete values are reported with the experiments in Section 6.

5.5. Two RL regimes

The central experimental contrast of the paper compares two ways of seeding the GRPO stage. *Late-init RL* starts from a saturated cold-start, that is, a checkpoint near the end of supervised training. *Early-init RL* starts from an early cold-start, before supervised training has converged. Both regimes share their reinforcement learning hyperparameters; only the initialization differs. We instantiate both regimes for all three cold-start regimes, yielding six reinforcement learning configurations.

5.6. Evaluation protocol

All evaluations use greedy decoding, verified bit-deterministic across reruns in Section 6.3. We report click-in-target accuracy on the in-distribution held-out WinDOM split, a deterministic 2,000-example subsample of the full 6,070-record test set fixed across all comparisons, and on three out-of-distribution real-screenshot benchmarks: ScreenSpot-Pro on professional desktop applications, OSWorld-G on open-domain operating-system tasks, and ScreenSpot-V2 on a broad mobile, web, and desktop mix, with respective sizes 1,530, 510, and 1,272. The unweighted mean of the three OOD scores is reported as *OOD-mean*.

6. Experiments

Throughout, the base model we train is **Qwen3.5-2B**, the ∼2B-parameter member of the Qwen3.5 vision-language family; every trained row in this paper (Direct SFT, both SFD modes, and all GRPO runs) starts from this same 2B base. The larger ∼4B same-family checkpoint appears *only* as an optional frozen teacher for the cross-size SFD mode (Section 5.3); it is never trained, never the deployed policy, and is not required by the recommended same-size EMA recipe. The full experimental setup, including backbone choice, tool-call schema and parser, training schedule, GPU choice, and evaluation decoding parameters, is deferred to Sections B to D. Section 6.1 reports the main results, Section 6.2 groups our three empirical findings, and Section 6.3 reports a sanity check on variance.

Table 1. Click-in-target accuracy (%). Each cold-start is followed by two GRPO variants: *+RL Early* initialises GRPO from step 100 of the cold-start, *+RL Late* from step 900. We report each row's best-OOD-mean checkpoint; OOD is the unweighted mean over SS-Pro, OSG, and SS-V2. **Bold** = best, underline = second among trained rows. Qwen3.5-4B is the cross-size SFD teacher, not trained here and not evaluated on SS-V2.

Model	In-dist.	SS-Pro	OSG	SS-V2	OOD
Qwen3.5-2B (base)	70.0	52.2	47.5	83.0	60.9
Qwen3.5-4B (base)	71.2	60.0	56.3	93.7	70.0
SFT	**82.1**	47.6	49.0	85.9	60.8
+RL Early	77.2	52.7	52.4	87.7	64.3
+RL Late	80.8	50.0	51.6	87.6	63.1
SFD-EMA	76.8	53.5	51.2	88.8	64.5
+RL Early	75.4	55.1	51.6	89.0	65.2
+RL Late	79.0	53.2	51.0	85.4	63.2
SFD-4B	76.5	54.7	50.8	**89.5**	65.0
+RL Early	76.5	**55.7**	**54.5**	88.8	**66.3**
+RL Late	79.7	53.0	51.4	88.9	64.4

6.1. Main results

Table 1 reports the best checkpoint per model on each benchmark; the Qwen3.5-2B and 4B bases are provided as references. Early-init RL on top of the cross-size SFD cold-start is the strongest of our nine configurations on the three-benchmark OOD mean: it leads on ScreenSpot-Pro and OSWorld-G among the trained rows, stays within one point of the strongest in-distribution row on ScreenSpot-V2, and over the Qwen3.5-2B base it gains +3.5 pp on ScreenSpot-Pro (55.7 vs 52.2), +7.0 pp on OSWorld-G (54.5 vs 47.5), +5.8 pp on ScreenSpot-V2 (88.8 vs 83.0), and +5.4 pp on the three-benchmark OOD-mean (66.3 vs 60.9). Figure 3 visualizes the same data as a scatter plot in the in-distribution / out-of-distribution plane, with marker shape encoding the training stage (circle: cold-start only, square: +RL Late, triangle: +RL Early) and marker colour encoding the cold-start type (blue: SFT, orange: SFD-EMA, magenta: SFD-4B): the Early-init RL family sits on the upper-left frontier, while the Late-init RL family clings to the cold-start curve.

6.2. Findings

F1: Supervised fine-tuning trades OOD accuracy for in-distribution. Direct SFT exhibits a clean memorization signature (Figure 4, top row): in-distribution accuracy rises from the Qwen3.5-2B base value of 70.0% to 82.1% over 1,000 steps (+12.1pp), while ScreenSpot-Pro accuracy

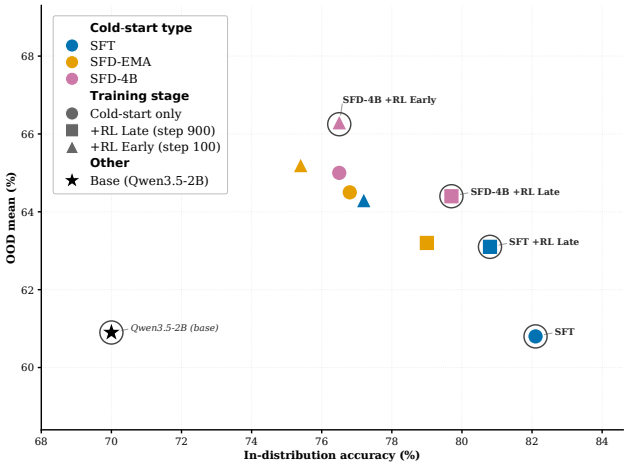

Figure 3. Best-checkpoint accuracy of every training run: in-distribution vs. OOD-mean (the three-benchmark mean of Table 1). Shape encodes stage (circle: cold-start, square: +RL Late, triangle: +RL Early); color encodes cold-start type (blue: SFT, orange: SFD-EMA, magenta: SFD-4B). The supervised cold-start, the late-init RL run, and our best configuration are highlighted.

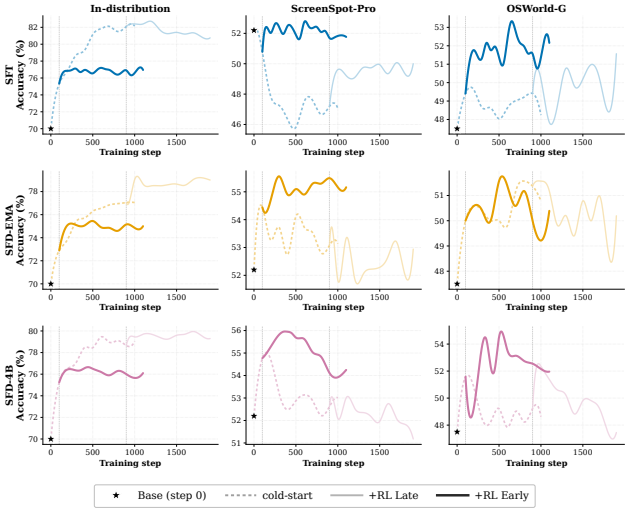

Figure 4. Click-in-target accuracy vs. training step. Rows: cold-start method (SFT, SFD-EMA, SFD-4B); columns: benchmark (in-distribution, ScreenSpot-Pro, OSWorld-G). Each cell shows the cold-start curve (gray dotted), the *+RL Late* path (gray solid, cold-start to step 900 then 1000 GRPO steps), and our best path *+RL Early* (accent colour, cold-start to step 100 then 1000 GRPO steps); vertical dashed lines mark the two GRPO start checkpoints. The SFT trap is visible in the top-row, middle-column panel (SFT $\times$ ScreenSpot-Pro): the SFT cold-start drops $\sim$4 points and the *+RL Late* path inherits that low operating point, whereas the SFD rows rise or stay flat and *+RL Early* climbs further. ScreenSpot-V2 trajectories are qualitatively similar and reported at each row's best checkpoint in Table 1.

decays from the base value of 52.2% down to 45.8% at step 500 and recovers only partially to 47.1% by step $1{,}000$ (-5.1pp net), with the worst ScreenSpot-Pro checkpoint sitting near the conventional one-epoch stopping point. The two effects almost exactly cancel on the OOD-mean axis: a fully converged SFT cold-start ends within a tenth of a point of the Qwen3.5-2B base (60.8 vs 60.9 in Table 1). Both Self-Family Distillation cold-starts soften the trade: ScreenSpot-Pro stays roughly flat across the cold-start phase rather than decaying, and the supervised stage already lifts the OOD-mean to 64.5 (SFD-EMA) and 65.0 (SFD-4B) before reinforcement learning begins.

F2: Cold-start depth controls RL transfer. The same GRPO recipe yields opposite OOD outcomes under the two cold-start depths. Late-init RL holds the in-distribution operating point in a narrow 79–81% band and barely moves ScreenSpot-Pro, but its OSWorld-G trajectory drifts downward: on the SFD-4B run, OSG enters GRPO at the cold-start value of 49.2, briefly rises to 52.4 at step 100, and then decays back to 47.5 at step $1{,}000$, the same value as the Qwen3.5-2B base, undoing both the cold-start gain and the brief in-RL bump.

Early-init RL takes the opposite trade. It gives up roughly 3–4 points of in-distribution accuracy and converts that headroom into clear OOD progress: on the same SFD-4B family, the Early-init trajectory peaks per-checkpoint at 55.9 on ScreenSpot-Pro and 54.5 on OSWorld-G (the headline row in Table 1 reports 55.7 and 54.5 at the best-OOD-mean checkpoint), and these peaks sit above *every* matched Late-init checkpoint we evaluate. The dominance is family-wide, not just at the peak: Early-init beats Late-init on three-benchmark OOD-mean at every GRPO step, by margins that grow consistently with cold-start quality: $63.1 \rightarrow 64.3$ for SFT ($+1.2$), $63.2 \rightarrow 65.2$ for SFD-EMA ($+2.0$), and $64.4 \rightarrow 66.3$ for SFD-4B ($+1.9$) (Figure 4).

The cold-start ledger pushes the same way. Late-init enters GRPO from a much longer cold-start (900 SFT steps vs Early-init's 100) and runs the same $1{,}000$-step GRPO phase on top, yet still loses on every OOD axis. Early-init therefore wins from a *less-trained* initialisation, a result that runs against the natural intuition that a more thoroughly converged cold-start should make it easier for GRPO to generalise out of distribution.

Remark (Hypothesis: entropy collapse). A natural account of why late-init RL fails to lift the OOD operating point is entropy collapse. A saturated cold-start has been driven by cross-entropy onto a near-deterministic action distribution: the per-prompt rollout group sampled from π_θ is concentrated on a single coordinate, so the within-group standard deviation σ in Equation (11) approaches zero and the group-relative advantage degenerates. Under that condition GRPO has no signal with which to refine OOD behaviors, since every rollout receives the same reward and contributes a zero-advantage update. The same pathology has been documented for clipped policy gradients in mathematical reasoning by Yu et al. (2025), and the broader observation that heavy supervised pre-training collapses the policy en-

tropy that subsequent RL relies on has been reported by DeepSeek-AI (2025); Chu et al. (2025); Shao et al. (2024). We have observed this signature in our training logs: the saturated SFD-4B cold-start enters GRPO with markedly lower per-token sampling entropy and a higher fraction of zero-variance rollout groups than the early cold-start, and the gap persists throughout the GRPO run. The empirical OOD contrast above is consistent with this account, though we do not claim it isolates entropy from alternative mechanisms (effective-update budget, on-policyness, distance from base); a quantitative entropy/variance plot across saturation levels is left to future work.

F3: Self-Family Distillation is robust to teacher choice. Both SFD modes track each other closely under Early-init RL: across every evaluated GRPO checkpoint, in-distribution accuracy stays in the 74.6 to 76.5% band and OOD-mean within ± 1.5 percentage points (Figure 4). The same-size EMA mode trails the cross-size 4B mode by roughly one OOD-mean point overall (65.2 vs 66.3): the two are essentially tied on ScreenSpot-V2 (89.0 vs 88.8), within 0.6 points on ScreenSpot-Pro (55.1 vs 55.7), and separated by 2.9 points on OSWorld-G (51.6 vs 54.5), where the residual gap concentrates. The practical takeaway is that the same-size mode, which needs no external model, is the recommended default; the cross-size mode is worth its extra compute only when a stronger same-family teacher is available cheaply. The two modes also differ in how often the teacher's rollout passes the click-in-bounding-box filter. With batch size 2, each training micro-batch contains two teacher rollouts; the micro-batch is kept and contributes a gradient if at least one of those two rollouts lands inside the GT bounding box, otherwise the entire micro-batch is discarded. Under that definition, 81.2% of micro-batches contribute gradients in the cross-size 4B mode and 62.1% in the same-size EMA mode (so 18.8% and 37.9% of micro-batches respectively are dropped at training time). The lower EMA kept-rate is consistent with the EMA teacher's lower per-step competence and confirms that the rejection-sampling filter remains a non-trivial verifier in either mode.

6.3. Variance

Eval determinism. Greedy decoding makes evaluation deterministic up to numerical noise. We reran the SFD-4B Early-init RL checkpoint at GRPO step 400 (the row that bolds OSWorld-G in Table 1) three times on different H100 GPUs. All three runs returned exactly $278/510 = 54.51\%$ on OSWorld-G with identical per-category breakdowns, and exactly $852/1530 = 55.69\%$ on ScreenSpot-Pro. The only source of variance in our reported numbers is therefore training itself, not evaluation.

Training variance. To bound the training-side noise of our headline result, we trained the SFD-4B Early-init RL configuration under three independent random seeds drawn at the trainer level, holding every other element of the recipe fixed: the cold-start checkpoint that initialises GRPO is byte-identical across the three runs, evaluation is greedy and therefore deterministic, and only the optimiser, data sampler, and rollout RNGs differ. The exact seed-propagation path through our training stack is described in Section I. On the best GRPO model the cross-seed accuracy range is 0.9pp on the in-distribution split, 0.6pp on ScreenSpot-Pro, and 0.4pp on OSWorld-G. The cross-seed spread is therefore smaller than the gap between SFD-4B Early-init RL and any Late-init RL or cold-start-only configuration in Table 1 on every benchmark we report, and we treat the headline numbers as effectively reproducible across seeds.

7. Discussion

We close by relating the recipe to safety properties for computer-use agents, then noting where the conclusions of Section 6 are most fragile and where the recipe is most likely to extend.

7.1. Provenance of the supervision channel

Every supervisory signal in the recipe, the rejection sampler at the SFD cold-start and the GRPO reward (Equation (10)), reduces to a single deterministic point-in-bounding-box check; no learned reward model, preference-data ranker, or LLM judge enters the loop. The cold-start labels are read from the renderer's own layout rather than from OCR or human raters, eliminating a class of label-noise failures (mislabelled crops, instruction–bbox swaps) at source. The resulting property is narrow: an outside reviewer can re-derive both the labels and the verifier from the released regeneration script, without trusting our compute environment. We make no corresponding claim about the behaviour of the trained policy at deployment.

7.2. Limitations

Our findings should be read against three scope restrictions.

Single backbone size: we evaluate one 2B backbone, leaving open whether the early-cold-start advantage transfers to the 7B and 72B regimes.

Single model family: both the student and the cross-size SFD teacher come from the Qwen3.5 family; we have not tested whether the same recipe works across families with different tokenisers or coordinate-emission protocols.

Static evaluation: our evaluation is single-click and static rather than agentic; we report transfer to real-screenshot grounding benchmarks but not to multi-step closed-loop environments.

7.3. Future work

Several directions follow naturally from the recipe and the artefact.

Online reinforcement learning. Because our pipeline includes both the dataset generator and the underlying renderer, WinDOM doubles as a fully autonomous environment: the same Win11React process that produces the static training corpus can serve on-policy rollouts at training time. The natural next step is to take GRPO online, using the renderer as a closed-loop reinforcement-learning environment that streams fresh, verifier-checked rollouts.

Online self-distillation. The same shift applies to the SFD cold-start: instead of pre-collecting a rejection-sampled corpus, an EMA self-teacher can generate supervision on the fly, with the click-in-bounding-box filter applied per-step.

Negative-rollout signal. The current rejection sampler discards every teacher rollout that misses the GT bounding box; with the deterministic verifier already in place, those rejected rollouts are a natural source of *negative* supervision for contrastive losses or preference-optimisation objectives such as DPO. Folding negatives into the cold-start would turn what is currently a wasted budget into a cheap form of preference data.

Long-horizon planning. Beyond single-click grounding, the same environment can host multi-step UI workflows that span several screens, opening a path from a single-click grounding agent to a planning agent within the same WinDOM training loop.

Verifier-only post-training beyond GUI grounding. The recipe (deterministic verifier, filtered same-family distillation, under-saturated cold-start) applies to any task that admits a programmatic ground-truth check, including refusal-format compliance, JSON-schema validity, code-execution unit tests, and math-answer equality; whether the cold-start-depth lever transfers to those tasks at 2B scale is an open empirical question.

Adversarial-UI evaluation. Scripting the same renderer with decoy DOM nodes, prompt-injected button labels, or spoofed accessibility roles converts Win11React from a training corpus into a closed-loop probe, which is the natural follow-up to the training-time-only safety claims of Section 7.4.

7.4. Broader Impact and Safety

Dual use. Reaching 55.7 / 54.5 / 88.8 on ScreenSpot-Pro, OSWorld-G, and ScreenSpot-V2 at 2B parameters and zero human bounding-box supervision lowers the absolute cost of building UI-driving agents, for both legitimate use (assistive desktop control, accessibility replacements, automated regression testing) and unattended abuse (credential stuffing, ad-fraud clickers). Withholding a 2B checkpoint does not meaningfully change this threat model relative to existing open GUI agents; the artefact reduces cost, not ceiling.

Auditability. Because WinDOM is regenerable from Win11React together with the released collection script, every $(\mathbf{x}, \mathcal{B}_{\text{GT}})$ pair is reconstructible by a third party, and the geometric verifier that scores it is deterministic. Section 6.3 additionally verifies bit-determinism of greedy evaluation across H100 GPUs on both OSWorld-G ($278/510$) and ScreenSpot-Pro ($852/1530$), so any reported accuracy can be independently re-derived from the released checkpoint. These are properties of the training and evaluation pipeline, not of the deployed policy.

Scope of the safety claim. What the design buys is supervision-channel auditability. We do not claim that the trained policy is safe to deploy unsupervised, robust to adversarial UI or prompt injection, or that the click-in-bounding-box reward functions as an alignment objective; an agent that clicks the right pixel can still be steered into harmful actions by the calling policy. Verifying any of those properties requires the dedicated experiments (adversarial-instruction evaluation, decoy-DOM probes, red-teaming inside the renderer) that we leave to future work.

8. Conclusion

Three takeaways follow from the experiments, and they together aim at one goal: making small grounding models work well on out-of-distribution screenshots. The browser DOM is a free, dense, layout-rectangle source of grounding labels, so any open-source web reimplementation of an operating system is a high-fidelity training corpus at zero annotation cost. In its same-size EMA mode, Self-Family Distillation provides a teacher-free cold-start that approaches the cross-size 4B-teacher variant within roughly one OOD-mean point (65.2 vs 66.3, Table 1) and lets small students avoid the cost of running a larger model alongside training. And cold-start depth is a hyperparameter that small models cannot ignore: the same RL recipe yields opposite OOD outcomes depending on whether the supervised base has been allowed to converge. Together they let a 2B vision-language model trained on synthetic Windows 11 data reach 88.8% on ScreenSpot-V2 and 55.7% on ScreenSpot-Pro without a single human bounding-box annotation.

Reproducibility Statement

We will release the code, training and evaluation scripts, the WinDOM corpus, the rendering pipeline, and the trained model weights under a permissive license at the camera-ready stage. All hyperparameters, system prompts, compute budgets, dataset details, full per-checkpoint accuracies, evaluation-determinism evidence, three-seed replication, and the headline-row reproduction recipe are reported in the appendix.

Upstream license and trade-dress. WinDOM is rendered against `blueedgetechno/win11React`, a community web reimplementation of Windows 11 released under the CC0-1.0 license, which permits redistribution of derivative renderings. The visual elements that the Win11React project itself replicates (window-control glyphs, the Start menu chrome, application icons, default Windows-style wallpapers) reproduce Microsoft trade-dress and are not licensed by Microsoft. To avoid redistributing trade-dress under our name we will, at camera-ready, (i) ship WinDOM as DOM snapshots, bounding boxes, and instructions plus a deterministic regeneration script that re-renders screenshots locally from Win11React rather than as a binary screenshot dump, and (ii) replace the curated wallpaper bundle with a permissively-licensed (CC0 / CC-BY) wallpaper set so that none of the redistributed image bytes originate from Microsoft assets. The exact license under which the regeneration script and the (instruction, bounding-box, DOM) tuples are released will be confirmed before camera-ready, and we will document any user-side caveats around running the regeneration pipeline on third-party trade-dress.

Broader Impact

Improved GUI grounding supports accessibility tools, automated software testing, and assistive desktop agents. The same capability enables misuse such as captcha bypass, scraping, and unauthorized account access. Our pipeline trains on a synthetic clone of Windows 11 and contains no personal data. Downstream agentic systems should add rate limiting, sandboxing, and human confirmation for destructive actions.

References

Chen, K., Shi, P., Qiu, H., Zeng, Z., Yang, S., Mao, W., and Ma, L. Metis-SPECS: Decoupling multimodal learning via self-distilled preference-based cold start. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=oNmMv7Lcj5.

Cheng, K., Sun, Q., Chu, Y., Xu, F., Li, Y., Zhang, J., and Wu, Z. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 9313–9332. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.505/.

Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/chu25c.html.

DeepSeek-AI. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1607–1616, 2018. URL https://proceedings.mlr.press/v80/furlanello18a.html.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Zhang, Y., Li, J., Xu, B., Dong, Y., Ding, M., and Tang, J. CogAgent: A visual language model for GUI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Hong_CogAgent_A_Visual_Language_Model_for_GUI_Agents_CVPR_2024_paper.pdf.

Hübotter, J., Lübeck, F., Behric, L., Baumann, A., Bagatella, M., Marta, D., Hakimi, I., Shenfeld, I., Buening, T. K., Guestrin, C., and Krause, A. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. URL https://arxiv.org/abs/2601.20802.

Jin, H., Luan, S., Lyu, S., Rabusseau, G., Rabbany, R., Precup, D., and Hamdaqa, M. RL fine-tuning heals OOD forgetting in SFT. *arXiv preprint arXiv:2509.12235*, 2025. URL https://arxiv.org/abs/2509.12235.

Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., Ma, J., Huang, Z., and Chua, T.-S. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM)*, pp. 8778–8786. ACM, 2025. doi: 10.1145/3746027.3755688. URL https://doi.org/10.1145/3746027.3755688.

Liu, Y., Li, P., Xie, C., Hu, X., Han, X., Zhang, S., Yang, H., and Wu, F. InfiGUI-R1: Advancing multimodal GUI agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*, 2025a. URL https://arxiv.org/abs/2504.14239.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding R1-Zero-Like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025b. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Lu, Z., Chai, Y., Guo, Y., Yin, X., Liu, L., Wang, H., Xiao, H., Ren, S., Xiong, G., and Li, H. UI-R1: Enhancing efficient action prediction of GUI agents by reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/38816.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Shenfeld, I., Damani, M., Hübotter, J., and Agrawal, P. Self-distillation enables continual learning. In *ICLR Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026. URL https://openreview.net/forum?id=HlWA3V6iKF.

Singh, A., Co-Reyes, J. D., Agarwal, R., Anand, A., et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=lNAyUngGFK.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html.

Wang, H., Zou, H., Song, H., et al. UI-TARS-2 technical report: Advancing GUI agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025. URL https://arxiv.org/abs/2509.02544.

Wu, Z., Wu, Z., Xu, F., Wang, Y., Sun, Q., Jia, C., Cheng, K., Ding, Z., Chen, L., Liang, P. P., and Qiao, Y. OS-Atlas: A foundation action model for generalist GUI agents. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=n9PDaFNi8t.

Xie, T., Deng, J., Li, X., Yang, J., Wu, H., Chen, J., Hu, W., Wang, X., Xu, Y., Wang, Z. A., Wang, Y., Yu, J., Yan, T., Zhao, C., and Yu, T. Scaling computer-use grounding via user interface decomposition and synthesis. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=FS0voKxELr.

Yang, Y., Li, D., Yang, Y., Luo, Z., Dai, Y., Chen, Z., Xu, R., Pan, L., Xiong, C., and Li, J. GTA1: GUI test-time scaling agent. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=3VIPmz7iAi.

Yu, Q., Zheng, Z., Song, S., et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. D. STaR: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022. URL https://openreview.net/forum?id=_3ELRdg2sgI.

Zhou, Y., Dai, S., Wang, S., Zhou, K., Jia, Q., and Xu, J. GUI-G1: Understanding R1-Zero-Like training for visual grounding in GUI agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=1XLjrmKZ4p.

A. Notation summary

This section consolidates every symbol, acronym, and named regime used in the paper into a single one-line glossary, so that the appendix can be read end to end without backward navigation.

Symbols.

- $\pi_\theta(\mathbf{o} \mid \mathbf{x})$: the student policy, a vision-language transformer that autoregressively samples an output sequence $\mathbf{o}$ given input $\mathbf{x}$ (a screenshot plus a textual instruction). See Section 3.2.
- π_{T}: the teacher policy. In same-size SFD, $\pi_{\mathrm{T}} = \pi_{\theta_{\text{EMA}}}$; in cross-size SFD, π_{T} is a frozen larger same-family checkpoint.
- π_{S}: the student policy under training. Used interchangeably with π_θ in figures.
- θ: the trainable parameters of the student.
- θ_{EMA}: the exponential moving average of the student parameters with decay α, updated as $\theta_{\text{EMA}} \leftarrow \alpha\, \theta_{\text{EMA}} + (1-\alpha)\, \theta$ on every optimizer step.
- α: EMA decay (we use $\alpha = 0.99$).
- $\mathbf{x}$: input to the policy (system prompt, screenshot, user instruction).
- $\mathbf{o} = (o_1, \ldots, o_T)$: output token sequence over the vocabulary $\mathcal{V}$.
- $\mathbf{y} \subseteq \mathbf{o}$: the structured action sub-sequence whose tokens encode the predicted click coordinate.
- $\mathbf{y}_{\text{teach}}, \mathbf{y}_{\text{student}}$: teacher and student action sub-sequences during distillation.
- $\mathrm{decode}(\cdot)$: a fixed parser $\mathcal{V}^* \to ([0,W]\times[0,H]) \cup \{\perp\}$ that returns either a click point in pixels or $\perp$ for non-parsable rollouts. $\perp$ counts as an incorrect click.
- $\mathcal{B}_{\text{GT}}$: ground-truth bounding box (axis-aligned rectangle) on the screenshot.
- (W, H): pixel dimensions of the screenshot. The model emits coordinates on a normalized $[0, 1000]^2$ grid that the parser rescales to (W, H).
- G: rollout group size in GRPO (we use $G = 8$).
- T_{rollout}: rollout sampling temperature (we use 1.3).
- $r(\mathbf{o}; \mathbf{x}, \mathcal{B}_{\text{GT}})$: verifiable reward, the sum of a 0.1 format bonus and a Gaussian click-in-target term (Equation (10)).
- $(\Delta x, \Delta y)$: pixel offset from predicted click to ground-truth box center.
- (σ_x, σ_y): half the GT-box width and height in pixels; sets the Gaussian reward bandwidth.
- $\hat{A}^{(i)}$: group-normalized advantage of the i-th rollout (Equation (11)).
- $\rho^{(i)}(\theta)$: importance ratio $\pi_\theta(\mathbf{o}^{(i)} \mid \mathbf{x}) / \pi_{\theta_{\text{old}}}(\mathbf{o}^{(i)} \mid \mathbf{x})$.
- ε: PPO-style clip range on $\rho^{(i)}$ (we use $\varepsilon = 0.2$).
- β: KL coefficient toward the cold-start policy in GRPO (we use $\beta = 0$).
- $\mathcal{L}_{\text{SFD}}$: forward-KL distillation loss at teacher-token positions (Equation (9)).
- $\mathcal{J}_{\text{GRPO}}$: clipped surrogate objective maximized in the RL stage.
- $\mathcal{D}$: training distribution over (screenshot, instruction, GT-box) triples; $\mathcal{D}_{\text{kept}}$ is the subset on which the teacher rollout passed the click-in-bbox filter.

Acronyms and named regimes.

- GUI: graphical user interface.
- DOM: document object model. The browser's tree of layout boxes; we read interactable bounding boxes from `getBoundingClientRect` on each DOM node.
- ARIA: accessible rich internet applications, the W3C accessibility role vocabulary used to filter interactable elements.
- SFT: supervised fine-tuning on (screenshot, instruction, action) tuples with the cross-entropy loss.
- SFD: Self-Family Distillation, our umbrella for two cold-start variants whose teacher is from the student's own model family.
- SFD-EMA: same-size SFD; the teacher is an EMA of the student.
- SFD-4B: cross-size SFD; the teacher is a frozen larger same-family checkpoint (in our instantiation a 4B parameter sibling of the 2B student).
- GRPO: Group Relative Policy Optimization; the RL algorithm of Section 5.4.
- KL: Kullback-Leibler divergence, used both as the SFD distillation loss and as an optional GRPO regularizer.
- Cold-start: the supervised checkpoint that initializes RL.
- Saturated cold-start: a cold-start whose in-distribution training-accuracy curve has plateaued (we report step 900 of 1,000).

- Early cold-start: a pre-plateau cold-start (we report step 100 of 1,000).
- Late-init RL: GRPO launched from a saturated cold-start.
- Early-init RL: GRPO launched from an early cold-start. The SFD-4B Early-init RL row is our headline configuration.
- In-distribution (in-dist.): a fixed 2,000-record subsample of the held-out WinDOM test split.
- OOD: out-of-distribution. We report three external benchmarks (ScreenSpot-Pro, OSWorld-G, ScreenSpot-V2) and their unweighted mean (OOD-mean).
- SS-Pro, OSG, SS-V2: shorthand column heads for ScreenSpot-Pro, OSWorld-G, and ScreenSpot-V2 in Table 1.
- Click-in-target accuracy: the fraction of evaluation instances where the predicted click point lies inside $\mathcal{B}_{\text{GT}}$ (Equation (5)). The only metric reported in this paper.
- WinDOM: our DOM-grounded dataset built from a community web reimplementation of Windows 11.
- QA-pass: the verification stage where a second-pass model classifies each candidate as keep, edit, or reject (Section 4).
- bbox: bounding box.
- pp: percentage points.

B. Experimental setup

This section gives the concrete instantiation and the full training schedule that lie behind every reported number. Together with the hyperparameters in Section C, the dataset construction in Section F, and the prompt schema in Section H, it suffices to reproduce a single training run from scratch on equivalent hardware.

Backbone choice and rationale. The base model trained in all experiments is **Qwen3.5-2B**, the ∼2B-parameter vision-language model in the Qwen3.5 family (Qwen Team, 2026); every trained checkpoint we report (Direct SFT, SFD-EMA, SFD-4B, and all GRPO runs) is this same 2B base, and the ∼4B same-family checkpoint is used only as an optional frozen teacher in the cross-size SFD mode, never as the trained or deployed model. We picked this family for three concrete reasons. First, it ships with a native tool-call interface (a JSON action block delimited by `<tool_call>...</tool_call>`) whose left-click action takes a coordinate on a normalized $[0, 1000]^2$ grid; this lets us reuse the model's pretrained tool-call vocabulary without adding an auxiliary grounding head, vocabulary expansion, or any custom modeling code. Second, a ∼4B-parameter same-family sibling shares the tokenizer and special-token table with the 2B student, which is exactly the alignment that the cross-size SFD loss Equation (9) requires (token-level forward KL only makes sense over a shared vocabulary). Third, both the 2B student and the 4B teacher are publicly available, run in standard `transformers` on a single H100 or H200 GPU, and produce the structured JSON actions that the parser below consumes. Replacing the backbone with a same-family pair from a different lineage (for example a ∼2B and a ∼4B from any other vision-language family with matching tokenizers and a documented coordinate emission convention) requires no change to the training recipe other than swapping the system prompt template (Section H).

Coordinate emission and parsing. At both training and evaluation time, the model is asked to emit a single tool call describing a left click. Concretely, the structured action is a JSON object with a tool name (`computer_use`) and an `arguments` field containing a `coordinate` entry. The `coordinate` entry is either a 2-tuple (x, y) in $[0, 1000]^2$ (a click point) or a 4-tuple (x_1, y_1, x_2, y_2) (a bounding-box prediction; we reduce it to its center for scoring). The parser (i) extracts everything between the first `<tool_call>` and the matching `</tool_call>`, (ii) parses it as JSON, (iii) reads `arguments.coordinate`, (iv) reduces a 4-tuple to its center, and (v) rescales the resulting integer point from the $[0, 1000]^2$ grid to absolute pixel coordinates by multiplying by $(W/1000, H/1000)$ where (W, H) is the screenshot resolution at evaluation time. Anything that fails to parse, or whose structured action is missing or malformed, returns $\perp$ and counts as an incorrect click. The same parser is shared by training-time reward computation (Equation (10)), evaluation-time accuracy (Equation (5)), and the click-in-bbox filter that gates teacher rollouts in SFD.

Decoding settings. All evaluations use greedy decoding: the model takes its argmax token at every position. We disable sampling, set the temperature to 1.0 (which is inert under argmax decoding), and disable top-k and top-p truncation. This makes the decoder bit-deterministic on a fixed GPU; small numerical differences across GPU types are discussed in Section E. The maximum new-token budget is 128 for the in-distribution split, OSWorld-G, and ScreenSpot-V2, and 1024 for ScreenSpot-Pro because the Pro instructions are noticeably longer and the model occasionally produces a longer chain of tool-internal text before emitting its action. At these budgets the parse-failure rate is below 0.1% on every benchmark we evaluate; we did not

observe a benchmark-by-benchmark sensitivity to the exact budget within $\pm 50\%$ of the values above.

Training schedule. The supervised cold-start runs for 1,000 optimizer steps in all three regimes (Direct SFT, SFD-EMA, SFD-4B), saving a checkpoint every 100 steps. We refer to the step-900 checkpoint as the *saturated* cold-start (its in-distribution training-accuracy curve has plateaued by then) and the step-100 checkpoint as the *early* cold-start. Both SFD modes present their teacher with the ground-truth bounding box jittered by ± 30 on the $[0, 1000]^2$ scale (independent integer noise per coordinate, clamped to $[0, 1000]$); the student is never shown the hint, only the teacher is. The same-size SFD-EMA mode uses parameter decay $\alpha = 0.99$, that is, $\theta_{\text{EMA}} \leftarrow 0.99\,\theta_{\text{EMA}} + 0.01\,\theta$ applied on every optimizer step. GRPO runs for an additional 1,000 steps starting from the chosen cold-start checkpoint, again saving every 100 steps; we report the GRPO checkpoint with the best mean accuracy across the three OOD benchmarks. Hardware: cold-start uses a single H200 GPU; GRPO and evaluation use a single H100 GPU. Wall-clock and total budgets are in Section D; the optimizer and reward hyperparameters are in Section C.

C. Hyperparameters

Tables 2 and 3 list every numeric hyperparameter used in cold-start and GRPO training. The same values were used for all rows of Table 1, with the single exception that the cross-size SFD teacher is absent in Direct SFT and SFD-EMA. Anything not listed (gradient clipping, weight decay, dropout) is left at the framework default for the underlying training library; we do not introduce dropout, weight tying, or auxiliary heads beyond the backbone's own.

Cold-start hyperparameter notes. The learning rate (5×10^{-6}) is at the low end of the typical SFT range for $\sim$2B vision-language models and was chosen to keep the SFD-EMA teacher (a moving average of the student) close to the student during the first epoch; raising the learning rate by an order of magnitude makes the EMA teacher chase a moving target and degrades F3 in our preliminary runs. The 1,000-step budget at effective batch size 32 corresponds to roughly 32,000 records, which is below one full epoch over the WinDOM training partition; we deliberately do not run multiple epochs because the SFT trap (Section 6.2) becomes more severe past one epoch.

GRPO hyperparameter notes. The group size $G = 8$ and the temperature $T_{\text{rollout}} = 1.3$ were chosen jointly: smaller groups make σ in Equation (11) estimate-noisy, and lower temperatures collapse the group onto a single coordinate. The completion-length cap of 64 tokens is enough for the longest valid tool-call action our parser has accepted in any of the runs we logged. The KL coefficient β is held at zero because, on our setup, it does not change the early-versus-saturated gap (see App. J); the cold-start policy enters only as the initialization of θ, not as a fixed reference. The reward weights (0.1 format + peak-1.0 Gaussian click-in-target) are calibrated so that a parsable but spatially incorrect rollout receives reward in $[0.1, \exp(-\frac{1}{2}) \cdot 1.0 + 0.1] \approx [0.1, 0.71]$ depending on its distance from the GT center, while a parsable click at the GT center receives 1.1 and a non-parsable rollout receives 0. This gradient is what allows GRPO to reward proximity rather than only success.

D. Compute

All training, including the cold-start and GRPO stages, runs on a single H200 GPU. All evaluation runs on a single H100 GPU. We did not use data or pipeline parallelism at any stage. The 2B student fits comfortably in a single H100 in bfloat16 at the listed batch size; the cross-size SFD setup additionally hosts the $\sim$4B teacher in inference mode in the same GPU memory, which is the constraint that pins the choice of teacher to a sibling no larger than $\sim$4B at this student size. We do not report exact wall-clock times per run, since they vary with preemption-induced restarts; we checkpoint every 100 steps so that a preemption costs at most one checkpoint of progress.

E. Evaluation determinism

Greedy decoding produces deterministic evaluation. Three reruns of our best configuration on OSWorld-G across different H100 GPUs returned identical accuracy (278/510 correct on each run). Partial-run accuracies on ScreenSpot-Pro can vary across reruns because evaluation-time sample order can differ, but the full-set accuracy is identical. We therefore attribute the variance reported in the main paper entirely to training stochasticity rather than evaluation noise.

F. WinDOM details

Per-scene randomization. Layout templates are sampled uniformly from {`single`, `corners`, `mosaic`, `cascading`, `split2`, `many_cascading`}. Viewports are drawn from the weighted pool {1920×1080: × 6, 2560×1440:×2, 1920×1200:×1, 1680×1050:×1}. Wallpapers are sampled uniformly from a curated bundle.

Bounding-box extraction. The interactable ARIA roles consumed by stage (iii) of Section 4 are: `button`, `link`, `textbox`, `menuitem`, `checkbox`, `radio`, `tab`, `switch`, `combobox`, `option`, `treeitem`, `listitem`, and `img`.

Table 2. Cold-start hyperparameters (shared across Direct SFT and both SFD modes).

Hyperparameter	Value
Student backbone	2B-parameter vision-language transformer (Section B)
Optimizer	AdamW (default β_1=0.9, β_2=0.999, ϵ=10^{-8})
Learning rate	5×10^{-6}
LR schedule	Cosine decay, 3% linear warm-up
Effective batch size	32 (gradient accumulation as needed for GPU memory)
Precision	bfloat16 (mixed precision; optimizer states in fp32)
Total steps	1,000 (one checkpoint every 100)
SFD-EMA decay α	0.99
SFD distillation loss	token-level forward KL at teacher-token positions (Equation (9))
SFD KL softmax temperature	1.0 (no softening of teacher logits)
Cross-size SFD teacher	frozen ∼4B same-family sibling of the student
Hint jitter (teacher-only)	± 30 integer noise per coordinate on the $[0, 1000]^2$ grid, clamped
Click-in-bbox filter	teacher rollout kept iff its parsed click lies inside $\mathcal{B}_{\text{GT}}$

Table 3. GRPO hyperparameters (shared across all six cold-start initializations).

Hyperparameter	Value
Group size G	8
Learning rate	2×10^{-6}
KL coefficient β (toward cold-start)	0.0
Clip range ε	0.2
Maximum completion length	64 tokens
Rollout sampling temperature T_{rollout}	1.3
Optimizer	AdamW (defaults as above)
Training prompts (subsample of WinDOM train)	8,000
Reward	$0.1 \cdot \mathbf{1}[\mathbf{o} \text{ parses}] + \exp\big(-\frac{1}{2}[(\Delta x/\sigma_x)^2 + (\Delta y/\sigma_y)^2]\big)$
(σ_x, σ_y)	half the GT bbox width and height in pixels
Total steps	1,000 (one checkpoint every 100)
Precision	bfloat16

Geometric cleaning. A box is dropped if its mean pixel brightness lies outside $[10, 235]$ on a $[0, 255]$ scale (blank or saturated regions), if its area exceeds 5% of the viewport (panels rather than elements), or if it would cause a trajectory to retain more than 36 elements. Perceptual-hash deduplication suppresses near-duplicate captures across trajectories.

Splits. Each trajectory identifier is bucketed by SHA-1 hash into target ratios of $85\%/5\%/10\%$ (train/val/test). Because trajectory record counts vary, the realized record-level shares are $82.9\%/6.0\%/11.2\%$ (45,095/3,260/6,070 records over $420/28/52$ trajectories). The held-out test split contains 6,070 records; the in-distribution evaluation in Section 6 uses a fixed 2,000-record subsample of that split for all comparisons.

Cleaned release. A separately maintained `cleaned-v2` release applies the geometric cleaning above more aggressively and is intended for downstream users that prefer fewer, higher-precision records; the experiments in this paper do not use it.

Provenance flags. Each record carries flags indicating which refinement stages it has passed (template, automatic refinement, automatic verification, vision-agent edit, human review), along with the layout template, viewport, parent scene, and developer-supplied action label. On the released 54,425-record corpus the QA-pass and edit fractions are: 100% (`_qa_full_pass`; the QA-pass classifier is run on every kept candidate), 25.8% (`_relabel_v2`; recovered after a Gemini-2.5-Flash relabel pass), 0.9% (`_agent_qa_edited`; edited by a Claude vision agent in the multi-agent QA pass), and 0.03% (`_human_corrected`; 19 records manually verified by a human reviewer). The keep/edit/reject ratios at the QA classifier stage are not retained per-record because rejected candidates were dropped at QA time rather than carried; we will publish the per-stage drop counts and the per-record provenance manifest with the corpus release so that downstream users can re-derive the keep/edit/reject distribution.

G. Headline-row reproduction recipe

The headline row of Table 1 (SFD-4B Early-init RL, $76.5/55.7/54.5/88.8$ on in-distribution / SS-Pro / OSG / SS-V2) is produced by the following three steps; every hyperparameter, prompt, and decoding setting referenced below is fixed in Sections B, C and H.

1. *Cold-start.* Train the SFD-4B mode of Section 5.3 on the WinDOM training partition for $1{,}000$ steps under the cold-start hyperparameters of Table 2, with the cross-size 4B teacher receiving the $\pm 30/[0, 1000]^2$ jittered ground-truth-bbox hint and the click-in-bounding-box rejection sampler of Section 5.3. Save the student checkpoint at supervised step 100.

2. *Reinforcement learning.* Initialise GRPO from the step-100 checkpoint and run for $1{,}000$ steps on $8{,}000$ training prompts subsampled deterministically from the WinDOM training partition under seed 42. Use the GRPO hyperparameters of Table 3, the Gaussian click-in-bounding-box reward of Equation (10), and the group-relative advantage of Equation (11). Save a checkpoint every 100 GRPO steps.

3. *Evaluation.* The checkpoint at GRPO step 400 is the headline row. Decode greedily under the protocol of Section B and score using the click-in-target indicator of Equation (5) on the in-distribution WinDOM split, ScreenSpot-Pro, OSWorld-G, and ScreenSpot-V2.

The cross-seed reproducibility of this row is reported in Section I.

H. System prompts and tool-call schema

The student model is trained and evaluated with the same system prompt, identical to the official Qwen-VL grounding template. Every input is a (system message, user message with image + textual instruction) pair, and the model is required to emit a single `<tool_call>...</tool_call>` JSON block describing a left-click. We reproduce the prompt verbatim:

```
You are a helpful assistant.  The
user will give you an instruction,
and you MUST left click on the
corresponding UI element via tool
call.  If you are not sure about
where to click, guess a most likely
one.
# Tools
You may call one or more functions
...resolution is {sw}x{sh}...
```

The placeholders `{sw}`, `{sh}` are filled in at runtime with the screenshot resolution. Both SFD teachers (cross-size 4B and same-size EMA self-teacher) receive the same system message but with an additional *hint* suffix appended to the user turn:

```
\nHint:  the target element is
approximately in the region [{hx1},
{hy1}, {hx2}, {hy2}] (xmin, ymin,
xmax, ymax).\nNow answer with the
tool call only:
```

The four hint coordinates are the ground-truth bounding box on the $[0, 1000]^2$ scale, jittered by an integer in $[-30, +30]$ per coordinate and clamped to range. The student is never shown this hint.

I. Three-seed GRPO replication

We launched three independent GRPO runs of our best configuration (Early-init RL on top of the SFD-4B cold-start at step 100), differing only in the seed passed to the trainer (`seed`$\in \{42, 123, 2024\}$). The seed is consumed in two places: (i) a `random.Random(seed)` that selects the $8{,}000$-prompt subsample from the WinDOM training partition, and (ii) the TRL `GRPOConfig.seed`, which Hugging Face Trainer forwards to `transformers.set_seed` (reseeding the Python `random`, NumPy, PyTorch CPU/CUDA RNGs, and the dataloader `RandomSampler` that controls prompt-shuffle order) and to GRPO's per-step rollout sampling RNG. The cold-start checkpoint loaded as the policy initialization is byte-identical across the three runs, and greedy evaluation is deterministic, so all reported between-seed variance is contributed by what the seed actually controls inside training. `seed=42` is the configuration reported in Table 1; all three seeds have completed evaluation across the in-distribution split, ScreenSpot-Pro, and OSWorld-G.

Cross-seed spread on the best model. On the best GRPO model the cross-seed accuracy range is 0.9pp on the in-distribution split, 0.6pp on ScreenSpot-Pro, and 0.4pp on OSWorld-G. The cross-seed gap between SFD-4B Early-init RL and any Late-init RL or cold-start-only configuration in Table 1 therefore exceeds the seed-induced spread on every benchmark.

J. What did not work

No-cold-start GRPO at 2B. Running GRPO directly on the base model failed: format-parse errors dominated the early reward signal and accuracy never exceeded thirty percent on the in-distribution split. At 2B scale, even a binary verifiable reward needs a warm-up. This is consistent with the larger-scale results reported in GTA1 (Yang et al., 2026) and points to a scale-dependent threshold.

Longer GRPO from saturated SFT. Extending Late-init RL with the SFT cold-start to twice the standard length left OOD accuracy unchanged and slowly degraded in-distribution accuracy. The late-init policy does not recover with additional GRPO steps; it is locked.

Mild KL regularization. Adding a small KL term toward the cold-start policy ($\beta = 0.01$) changed the early-versus-saturated gap by less than one point on ScreenSpot-Pro. The transfer gap originates in the cold-start, not in KL drift during reinforcement learning.