CoEvolve: Construct-to-Edit Visual Grounding with Bidirectional State Refinement

Anonymous authors
Paper under double-blind review

Abstract

Visual grounding localizes an object described by language with a bounding box. Most multimodal grounding models compress target identification, spatial reasoning, and boundary estimation into one terminal prediction. Free-form rationales make reasoning linguistically explicit but do not necessarily expose measurable, editable spatial states. Intermediate localization errors are therefore difficult to diagnose and correct, allowing incorrect region choices and imprecise boundaries to persist in the final box. We introduce **CoEvolve**, a construct-to-edit framework that separates grounding into explicit state construction and state editing. Region-Evolution Reinforcement (RER) organizes grounding analysis into a progressive semantic–spatial trajectory, with each reasoning step committing to an explicit candidate region. Bidirectional Denoising Refiner (BDR) treats the reasoning text as fixed semantic context and refines the trajectory's coordinate fields through bidirectional same-position reconstruction. Geometry- and behavior-level objectives provide target geometry and edit-preference signals for consolidating reliable candidates, preserving accurate inputs, or correcting toward annotations. Evaluations cover natural-image and remote-sensing grounding. With a 9B backbone, CoEvolve rivals models up to 241B parameters in grounding accuracy. Under controlled corruption, a single BDR pass improves mean box overlap by over 27 percentage points, demonstrating strong recovery from substantial localization errors. State-source comparisons further support the complementarity of explicit state construction and source-matched editing.

1 Introduction

Visual grounding, or referring expression comprehension, localizes a language-described object with a bounding box. The field has progressed from proposal-based and modular grounding (Kazemzadeh et al., 2014; Mao et al., 2016; Yu et al., 2016; 2018; Zhang et al., 2018; Liu et al., 2019b; Yang et al., 2019; Liu et al., 2019a) to end-to-end Transformers and grounded pre-training (Deng et al., 2021; Kamath et al., 2021; Li et al., 2022; Yan et al., 2023; Liu et al., 2024; Wang et al., 2023), and more recently to multimodal large language models (MLLMs) that unify language understanding and coordinate generation (You et al., 2024; Wang et al., 2024a;b; Bai et al., 2025a;b; Wang et al., 2025; Wu et al., 2024). Despite stronger semantic representations, most approaches still compress target disambiguation, spatial search, and boundary estimation into one terminal box. Without an explicit spatial state for inspection and repair, errors in region selection and boundary estimation can persist into the final prediction. Reasoning-oriented systems expose rationales before spatial prediction (DeepSeek-AI, 2025; Huang et al., 2026b; Liu et al., 2025), and grounded reasoning methods further pair language with explicit visual regions or coordinates (Fan et al., 2025; Wu et al., 2026). Iterative localization has also used reinforcement-guided region search (Sun et al., 2021), including anchor-guided reasoning in remote sensing (Wang et al., 2026a). These advances motivate a further question: how can candidate target regions form a measurable state that is passed to an editor, so that coordinate changes can be evaluated against the observed state and accurate regions can be preserved?

We introduce **CoEvolve**, a construct-to-edit framework that separates grounding into explicit state construction and state editing, as illustrated in Fig. 1. **Region-Evolution Reinforcement (RER)**

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.

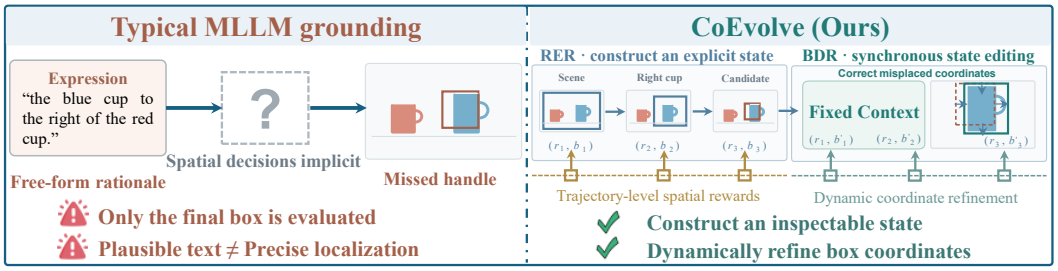

Figure 1: Comparison between conventional MLLM grounding and CoEvolve. Conventional methods compress spatial reasoning and localization into a terminal prediction, leaving intermediate spatial decisions implicit and difficult to correct. CoEvolve uses RER to construct a measurable progressive region trajectory and BDR to synchronously refine its coordinate fields while retaining the surrounding response structure.

externalizes otherwise free-form grounding analysis into a progressive semantic–spatial trajectory in which every reasoning step commits to an explicit region. Trajectory-level spatial rewards derived from terminal annotations adapt the model to this structured interface without human-authored reasoning chains or intermediate region labels. **Bidirectional Denoising Refiner (BDR)** retains the reasoning text as semantic context and edits the trajectory's coordinate fields through bidirectional same-position reconstruction. Geometry-level supervision provides target geometry, while behavior-level supervision provides preferences for consolidating a reliable candidate, preserving an accurate input, or correcting toward the annotation.

Experiments evaluate localization accuracy and state refinement. CoEvolve reaches 92.41% Overall Acc@0.5 on eight RefCOCO-family splits and 85.72% mIoU on five test splits. Under controlled corruption, BDR raises mIoU from 58.80% to 86.19%, with 87.99% Repair@0.7. Domain-specific retraining achieves 76.23% mIoU on DIOR-RSVG. Our contributions are:

- We introduce CoEvolve and its RER component to expose intermediate localization decisions as a parseable semantic–spatial trajectory, providing an explicit state for inspection and correction.
- We develop BDR to edit the coordinate fields of this state through position-aligned same-position reconstruction. Geometry- and behavior-level objectives jointly provide target geometry and edit-preference signals for state refinement.
- We demonstrate strong target localization across natural-image and remote-sensing benchmarks, together with substantial correction gains under controlled box corruption.

2 Preliminaries

2.1 Group Relative Policy Optimization

Let $c = (I, q)$ denote a multimodal context consisting of an image I and a referring expression q, and let $y = (y_1, \ldots, y_T)$ be a response containing the reasoning trajectory and final coordinates. An autoregressive policy factorizes the response likelihood as

$$\pi_\theta(y \mid c) = \prod_{t=1}^{T} \pi_\theta(y_t \mid c, y_{<t}). \tag{1}$$

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) removes the learned critic by sampling a group $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid c)$ and normalizing sequence-level rewards R_i within the group:

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \delta}, \qquad \mu_R = \frac{1}{G}\sum_{i=1}^{G} R_i, \tag{2}$$

where σ_R is the corresponding standard deviation and δ ensures numerical stability. With the token-wise importance ratio

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid c, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid c, y_{i,<t})},$$

the policy is optimized by

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\Big\{\rho_{i,t}(\theta)A_i, \\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon_l, 1+\epsilon_h)\,A_i\Big\} - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Bigg]. \tag{3}$$

Here π_{ref} is the reference policy, β controls KL regularization, and ϵ_l and ϵ_h are the lower and upper clipping radii. Setting $\epsilon_l = \epsilon_h$ recovers symmetric clipping; using $\epsilon_h > \epsilon_l$ permits broader exploration for positively advantaged responses.

2.2 Bidirectional Denoising Refinement

Bidirectional reconstruction over corrupted discrete tokens provides a natural bridge between bidirectional context modeling and iterative correction (Ou et al., 2025; Sahoo et al., 2024; Sun et al., 2026). Lavida-O applies masked diffusion with bidirectional context to discretized grounding coordinates (Li et al., 2026). BDR instead keeps an existing response and its coordinate tokens visible as references for same-position editing. Let $z = (z_1, \ldots, z_L)$ denote a clean response and $\widetilde{z}$ its discretely corrupted version, with $\mathcal{M}$ denoting the perturbed positions. A bidirectional denoiser reconstructs the original token at each selected position with the same-position objective:

$$\mathcal{L}_{\text{denoise}} = -\mathbb{E}_{z,\widetilde{z}}\left[\frac{1}{|\mathcal{M}|}\sum_{j\in\mathcal{M}}\log p_\phi(z_j \mid \widetilde{z}, c)_j\right]. \tag{4}$$

Here $p_\phi(\cdot \mid \widetilde{z}, c)_j$ is the token distribution predicted at response position j. Starting from an imperfect response $z^{(0)}$, the same position-aligned predictor can update selected positions for P passes:

$$z_j^{(p)} = \begin{cases} \arg\max_v \; p_\phi\Big(v \mid z^{(p-1)}, c\Big)_j, & j \in \mathcal{M}^{(p)}, \\ z_j^{(p-1)}, & \text{otherwise}, \end{cases} \qquad p = 1, \ldots, P. \tag{5}$$

Here $\mathcal{M}^{(p)}$ denotes the set of response positions updated during refinement pass p.

This formulation combines bidirectional position-aligned reconstruction with denoising-style iterative correction. It operates directly on discrete response tokens rather than introducing a continuous noise-time process or a full reverse-time Markov chain. In CoEvolve, $z^{(0)}$ is the RER localization trajectory, and BDR extends this view to visible and geometrically corrupted coordinate tokens; token masking is used only during warm-up. Bidirectional context and coordinate-aware objectives define the refinement process described in Section 3.

3 Methodology

3.1 Overview

Given an image I and a referring expression q, visual grounding predicts the referred object's box in the non-empty target setting. We denote the training annotation by $\mathbf{b}^* = [x_1^*, y_1^*, x_2^*, y_2^*]$ and normalize coordinates to integers in $[0, 1000]$. A valid box satisfies $x_1 < x_2$ and $y_1 < y_2$ within this range. Fig. 2 illustrates the workflow. **Region-Evolution Reinforcement (RER)** constructs a tagged response y comprising reasoning text, a progressive region sequence, and a terminal answer. **Bidirectional Denoising Refiner (BDR)** retains the reasoning text as semantic context and refines the state's coordinate fields. This state provides semantic context, spatial history, an edit reference, and natural upstream errors. The two components are trained sequentially and communicate only through the generated state.

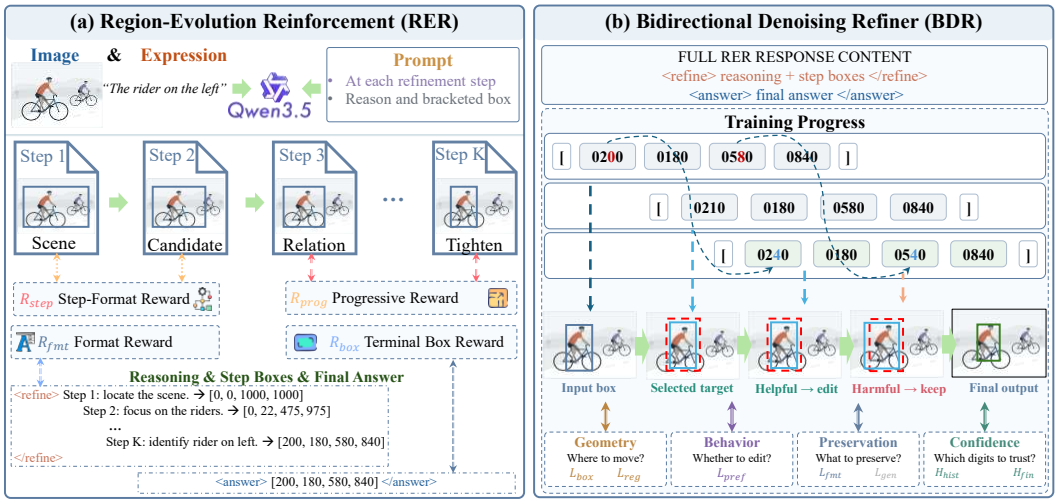

Figure 2: Overview of CoEvolve. RER converts otherwise free-form grounding analysis into a progressive semantic–spatial state in which each reasoning step commits to an explicit region. The resulting response becomes an editable localization state for BDR. BDR performs bidirectional same-position refinement of its coordinate fields while retaining the surrounding response as context. Geometry-level objectives provide target geometry, and behavior-level objectives provide edit-preference signals.

3.2 Region-Evolution Reinforcement

An outcome-only box reward does not directly assess whether intermediate region states make target-directed progress before the terminal prediction. RER therefore adapts an autoregressive policy $\pi_\theta(y \mid I, q)$ to construct a measurable localization state. A valid response contains one `<refine>` block with 3–5 steps and one `<answer>` block. We denote its parsed sequence by $\tau(y)$, where each pair $(r_k, \mathbf{b}_k)$ associates a reasoning step with an explicit region and $\mathbf{b}_{\text{ans}}$ is the terminal answer. Unlike regions that identify visual evidence in grounded reasoning (Fan et al., 2025; Wu et al., 2026), each RER box is a candidate location for the referred target, and the ordered sequence becomes BDR's editable state. We use *semantic–spatial state* to denote this division of roles: free-form text supplies semantic context, while each step exposes a spatial commitment evaluated by trajectory-level rewards. These rewards assess the spatial commitments rather than the semantic correctness of the free-form text itself. For each sampled response, RER assigns the sequence-level reward

$$R_{\text{RER}}(y, \mathbf{b}^*) = 0.5R_{\text{fmt}}(y) + 0.5R_{\text{step}}(y) + R_{\text{prog}}(y, \mathbf{b}^*) + R_{\text{box}}(y, \mathbf{b}^*). \tag{6}$$

R_{fmt} checks the outer tagged structure and provides graded feedback for partially valid responses. R_{step} checks whether each refinement step ends in a parseable box; it does not score localization accuracy. When 3–5 valid non-empty regions are available, R_{prog} evaluates target-directed spatial progress using overlap, center, coverage, and final-consistency signals. R_{box} parses and evaluates the terminal answer by its overlap and shape agreement with $\mathbf{b}^*$, and assigns the prescribed negative reward when the answer is missing, repeated, malformed, or invalid. The appendix specifies the piecewise definitions and coefficients of these terms.

We sample eight responses per input and use Eq. (6) in the GRPO objective introduced in Section 2. The annotated terminal box evaluates the generated regions, but RER requires no manually labeled reasoning text or intermediate box trajectory. It therefore derives trajectory-level spatial rewards from the terminal box annotation and organizes the search process into a structured state whose coordinate fields BDR can refine.

3.3 Bidirectional Denoising Refiner

Editable localization states. BDR refines complete RER responses, using their fixed reasoning text as semantic context and editing the coordinate fields of historical Steps, the last Step, and

the Answer. Geometry-level and behavior-level objectives supervise target coordinates and edit decisions, while auxiliary reconstruction and entropy terms preserve the response and regularize confidence. For states whose RER Answer has IoU above 0.5, we sample $\xi \sim \mathrm{Bernoulli}(0.3)$; for all other states, we set $\xi = 0$. Given $\mathbf{x}^{(0)} = \mathrm{Tok}(y)$, we use $\mathbf{x}^{\mathrm{in}} = \mathbf{x}^{(0)}$ when $\xi = 0$ and $\mathbf{x}^{\mathrm{in}} = \mathrm{Tok}(\mathcal{Q}(y, \mathbf{b}^*))$ otherwise, where $\mathcal{Q}$ replaces selected boxes with valid geometric perturbations.

Bidirectional same-position reconstruction. We realize bidirectional state access by removing causal masking from the backbone's full-attention layers, while retaining each editable coordinate at its original response position. The set $\mathcal{T}_s$ contains the 16 ordered digit positions of box role s, and $\mathcal{T}_{\mathrm{bbox}} = \bigcup_s \mathcal{T}_s$. BDR produces position-aligned logits for all editable coordinates from the same observed response. Denoting the ten digit tokens by $\mathcal{V}_{\mathrm{digit}}$, the coordinate candidates used during training are obtained as

$$\begin{aligned} \mathbf{z} = F_\phi(I, q, \mathbf{x}^{\mathrm{in}}), \quad \tilde{x}_t^{\mathrm{out}} = \arg\max_{v \in \mathcal{V}_{\mathrm{digit}}} z_{t,v}, \quad t \in \mathcal{T}_{\mathrm{bbox}}, \\ \tilde{\mathbf{b}}_s^{\mathrm{out}} = \mathrm{DecBox}\big((\tilde{x}_t^{\mathrm{out}})_{t \in \mathcal{T}_s}\big). \end{aligned} \tag{7}$$

Here F_ϕ denotes BDR with adapter parameters ϕ, and $\mathbf{z}$ contains position-aligned logits. Digit projection yields synchronous box candidates for target selection, regression, and preference supervision. At inference, BDR updates coordinate positions while retaining the reasoning text; denoising denotes reconstruction from masking and geometric corruption, not a diffusion timestep process.

Behavior-Level Supervision: Quality-Gated Target Selection. We compute $u_{\mathrm{in}} = \mathrm{IoU}(\mathbf{b}_{\mathrm{ans}}^{\mathrm{in}}, \mathbf{b}^*)$ and $u_{\mathrm{out}} = \mathrm{IoU}(\tilde{\mathbf{b}}_{\mathrm{ans}}^{\mathrm{out}}, \mathbf{b}^*)$, assigning zero to an invalid box. With $\eta = 0.9$ and $\gamma = 0.005$, $\mathcal{C}_{\mathrm{out}}$ denotes $u_{\mathrm{out}} > \eta$ and $u_{\mathrm{out}} - u_{\mathrm{in}} \geq \gamma$, while $\mathcal{C}_{\mathrm{in}}$ denotes $\xi = 0$ and $u_{\mathrm{in}} \geq \eta$. Absolute box supervision uses

$$\mathbf{b}^{\mathrm{tar}} = \begin{cases} \tilde{\mathbf{b}}_{\mathrm{ans}}^{\mathrm{out}}, & \mathcal{C}_{\mathrm{out}}, \\ \mathbf{b}_{\mathrm{ans}}^{\mathrm{in}}, & \mathcal{C}_{\mathrm{in}}, \\ \mathbf{b}^*, & \text{otherwise}. \end{cases} \tag{8}$$

The stop-gradient branches define three target-selection cases. When $\mathcal{C}_{\mathrm{out}}$ holds, the candidate itself becomes the target, reinforcing a reliable improvement. When $\mathcal{C}_{\mathrm{in}}$ holds, the input anchor becomes the target, reinforcing preservation. Otherwise, the annotation provides the corrective target. The rule therefore supplies both the geometric label and the corresponding training preference for the current state.

Geometry-Level Supervision. Geometry-level supervision provides the target geometry for the edited localization. The last refinement box and the answer, denoted by $\mathcal{S}_f = \{\mathrm{last}, \mathrm{ans}\}$, receive the same selected target from Eq. (8). Their four zero-padded coordinates contribute 32 strongly supervised digit positions. The coordinate cross-entropy $\mathcal{L}_{\mathrm{box}}$ enforces the target digit at each position, with equal digit weights during warm-up. After warm-up, the four digits of coordinate k share $w_k = 1 + 10e_k$, where $e_k = (2000)^{-1} \sum_{s \in \mathcal{S}_f} |\tilde{b}_{sk}^{\mathrm{out}} - b_k^{\mathrm{tar}}|$. The weighted token losses are divided by the sum of their weights, so w_k redistributes gradient toward inaccurate coordinates without changing the nominal scale of $\mathcal{L}_{\mathrm{box}}$.

Token cross-entropy distinguishes exact digit identities but does not directly encode numerical distance. We therefore project the logits of digit $j \in \{0, 1, 2, 3\}$ in coordinate $k \in \{1, 2, 3, 4\}$ of role $s \in \mathcal{S}_f$ onto the ten digit tokens. For its position t_{skj}, the resulting distribution and soft coordinate are

$$p_{skj}(d) = \frac{\exp z_{t_{skj},d}}{\sum_{a=0}^{9} \exp z_{t_{skj},a}}, \bar{b}_{sk} = \sum_{j=0}^{3} 10^{3-j} \sum_{d=0}^{9} d\, p_{skj}(d). \tag{9}$$

The regression loss $\mathcal{L}_{\mathrm{reg}}$ averages $|\bar{b}_{sk} - b_k^{\mathrm{tar}}|/1000$ over the eight supervised coordinates. Together, $\mathcal{L}_{\mathrm{box}}$ and $\mathcal{L}_{\mathrm{reg}}$ provide complementary geometric signals: the former preserves the exact discrete coordinate representation, while the latter encodes continuous numerical distance to the selected target.

Coordinate-wise Edit Preference. Quality-gated target selection defines the desired state-level behavior, but does not explicitly compare an individual coordinate edit with retaining its input value.

We therefore introduce coordinate-wise preference supervision. We index historical roles by $\mathcal{S}_h = \{1, \ldots, K-1\}$ and define $\mathcal{S} = \mathcal{S}_h \cup \mathcal{S}_f$. For every role with 16 valid input digits, $\breve{\mathbf{b}}_s^{\text{in}}$ is decoded from the observed input and $\tilde{\mathbf{b}}_s^{\text{out}}$ is the digit-projected candidate. Their coordinate-level edit gain is

$$\Delta_{sk} = \frac{|\breve{b}_{sk}^{\text{in}} - b_k^*| - |\tilde{b}_{sk}^{\text{out}} - b_k^*|}{1000}. \tag{10}$$

For coordinate k of role s, $\mathcal{C}_{sk}$ contains the changed digit positions, D_{sk} is their mean output-versus-input logit margin, and y_{sk} indicates whether Δ_{sk} is positive. Thus, $\Delta_{sk} > 0$ identifies an edit closer to the annotation than its input counterpart, whereas $\Delta_{sk} < 0$ identifies a detrimental edit. Binary cross-entropy with logits is applied when $\mathcal{C}_{sk} \neq \varnothing$ and $|\Delta_{sk}| > 0.005$. Active coordinates within a role are averaged with weights $a_{sk} = \min(1, |\Delta_{sk}|/0.05)$. Historical roles receive weight 0.2, and the last Step and Answer each receive weight 0.5. The active role losses are summed per sample and then averaged over the minibatch to obtain $\mathcal{L}_{\text{pref}}$. Unlike absolute coordinate supervision, $\mathcal{L}_{\text{pref}}$ learns the relative utility of an edit.

State Preservation and Confidence Regularization. The primary objectives operate inside a structured response whose reasoning and syntax must remain stable. General reconstruction $\mathcal{L}_{\text{gen}}$ applies token cross-entropy outside the strongly supervised final digits; after warm-up, it preserves non-coordinate assistant content. Format loss $\mathcal{L}_{\text{fmt}}$ adds cross-entropy on structural tags and coordinate punctuation. For a digit distribution p, natural entropy is $H(p) = -\sum_{d=0}^{9} p(d) \log p(d)$. The historical term $\mathcal{H}_{\text{hist}}$ sharpens digit distributions in earlier refinement steps without imposing an absolute coordinate target. The final term $\mathcal{H}_{\text{fin}}$ acts on the last Step and Answer only when the digit-vocabulary argmax equals the selected target digit, sharpening correct predictions without increasing confidence in an incorrect digit. These terms preserve response structure and non-coordinate content while regularizing confidence; they do not determine whether an edit is beneficial.

Combined objective and warm-up. With the coefficients used in training, the post-warm-up objective is

$$\begin{aligned}\mathcal{L}_{\text{BDR}} = \underbrace{\mathcal{L}_{\text{box}} + 30\mathcal{L}_{\text{reg}}}_{\mathcal{L}_{\text{geom}}} + \mathcal{L}_{\text{pref}} \\ + \underbrace{0.5\mathcal{L}_{\text{fmt}} + 0.2\mathcal{L}_{\text{gen}}}_{\mathcal{L}_{\text{state}}} + \underbrace{0.2\mathcal{H}_{\text{hist}} + 0.2\mathcal{H}_{\text{fin}}}_{\mathcal{L}_{\text{conf}}}.\end{aligned} \tag{11}$$

Behavior-level supervision enters through both the target policy in Eq. (8), which determines the target used by $\mathcal{L}_{\text{geom}}$, and the additive preference term $\mathcal{L}_{\text{pref}}$. During the first 3% of optimization, we mask 10% of eligible assistant tokens and use $\mathcal{L}_{\text{box}} + 0.2\mathcal{L}_{\text{gen}}$. The target rule remains active; $\mathcal{L}_{\text{gen}}$ also covers historical box digits, assigning the selected target to perturbed historical boxes. Regression, entropy, and preference are activated after warm-up.

4 Experiments

We evaluate CoEvolve on standard referring-expression grounding, localization under stricter IoU criteria, controlled box correction, component ablation, and remote-sensing grounding. Unless otherwise noted, aggregate evaluation metrics are reported as percentages.

4.1 Experimental Setup

Datasets and Evaluation. For natural-image grounding, we train CoEvolve only on the RefCOCOg training set and evaluate all eight RefCOCO-family splits (57,457 examples) without further fine-tuning (Mao et al., 2016; Yu et al., 2016). RefCOCOg measures in-dataset performance, while RefCOCO and RefCOCO+ provide cross-dataset evaluation under RefCOCOg-only training. We report split-wise $\text{Acc@}0.5$ and their arithmetic mean (Overall). Stricter-IoU, controlled-correction, and ablation experiments use five test splits comprising 30,969 examples. Unless stated otherwise, their aggregate metrics are computed jointly over all examples rather than by averaging the five split-level scores. Remote-sensing grounding is evaluated on DIOR-RSVG (Zhan et al., 2023).

4.2 STANDARD REFCOCO-FAMILY GROUNDING

Among the general-purpose MLLMs, specialized grounding systems, and grounding-adapted MLLMs compared in Table 1, CoEvolve obtains the highest tabulated Overall Acc@0.5 of 92.41, outperforming Qwen3.5-9B on all eight splits and improving its aggregate score by 2.93 points. CoEvolve ranks first on four splits and ties for first on RefCOCOg val, demonstrating strong and consistent grounding performance across the RefCOCO family.

Table 1: Grounding performance on the eight standard RefCOCO-family splits, measured by Acc@0.5. Overall is the arithmetic mean across the eight split-level results. It is computed from full-precision values when available and otherwise from the reported split-level values. Public methods follow their reported protocols; ‡ denotes our locally reproduced Qwen3.5-9B baseline. CoEvolve trains only on RefCOCOg train, without target-dataset fine-tuning. Best and second-best results are bolded and underlined.

	RefCOCO			RefCOCO+			RefCOCOg		
Method	val	testA	testB	val	testA	testB	val	test	Overall ↑
Qwen2-VL-7B (Wang et al., 2024a)	91.7	93.6	87.3	85.8	90.5	79.5	87.3	87.8	87.94
Qwen2-VL-72B (Wang et al., 2024a)	93.2	95.3	90.7	90.1	<u>93.8</u>	85.6	89.9	90.4	91.13
Qwen2.5-VL-7B (Bai et al., 2025b)	90.0	92.5	85.4	84.2	89.1	76.9	87.2	87.2	86.56
Qwen2.5-VL-72B (Bai et al., 2025b)	92.7	94.6	89.7	88.9	92.2	83.7	89.9	90.3	90.25
Qwen3-VL-8B-Instruct (Bai et al., 2025a)	91.6	93.3	87.8	85.8	90.3	79.9	88.7	88.7	88.26
Qwen3-VL-235B-A22B (Bai et al., 2025a)	92.9	94.6	90.2	88.6	92.4	84.2	90.3	90.8	90.50
InternVL3-8B (Zhu et al., 2025)	92.5	94.6	88.0	88.2	92.5	81.8	89.6	90.0	89.65
InternVL3-78B (Zhu et al., 2025)	93.4	<u>95.4</u>	90.3	90.1	<u>93.8</u>	85.3	91.5	91.5	91.41
InternVL3.5-8B (Wang et al., 2025)	92.4	94.7	88.7	87.9	92.4	82.4	89.6	89.4	89.69
InternVL3.5-241B-A28B (Wang et al., 2025)	<u>94.1</u>	**96.3**	<u>91.5</u>	**91.6**	**94.6**	<u>86.9</u>	<u>92.0</u>	92.1	<u>92.39</u>
DeepSeek-VL2-Small (Wu et al., 2024)	93.9	95.3	91.3	89.4	92.9	84.8	**92.6**	<u>92.6</u>	91.60
Grounding-DINO-L (Liu et al., 2024)	90.6	93.2	88.2	82.8	89.0	75.9	86.1	87.0	86.60
UNINEXT-H (Yan et al., 2023)	92.6	94.3	<u>91.5</u>	85.2	89.6	79.8	88.7	89.4	88.89
ONE-PEACE (Wang et al., 2023)	92.6	94.2	89.3	88.8	92.2	83.2	89.2	89.3	89.85
Ferret-v2-13B (Zhang et al., 2024)	92.6	95.0	88.9	87.4	92.1	81.4	89.4	90.0	89.60
CogVLM-Grounding-17B (Wang et al., 2024b)	92.8	94.8	89.0	88.7	92.9	83.4	89.8	90.8	90.28
EGM-InternVL3-8B (Zhan et al., 2026)	93.6	95.2	90.1	89.3	93.6	83.1	89.7	90.7	90.66
EGM-Qwen3-VL-8B (Zhan et al., 2026)	93.9	95.0	91.2	90.1	93.3	85.9	90.4	91.2	91.38
Qwen3.5-9B‡ (Qwen Team, 2026)	91.5	93.8	89.5	87.7	91.3	82.9	89.7	89.6	89.48
CoEvolve (ours)	**94.4**	95.3	**92.4**	<u>90.7</u>	93.5	**87.8**	**92.6**	**92.8**	**92.41**

4.3 LOCALIZATION UNDER STRICTER IOU CRITERIA

Table 2 evaluates localization under stringent IoU criteria across five test splits. CoEvolve achieves 85.72% mIoU, 87.94% Acc@0.7, and 68.64% Acc@0.9, exceeding CogVLM-Grounding-17B by 2.12, 2.63, and 4.19 points, respectively. CoEvolve ranks first in every reported setting, with Acc@0.9 gains ranging from 0.41 points on RefCOCOg test to 7.02 points on RefCOCO+ testB. The larger gains at the strictest threshold are consistent with improved high-IoU localization and boundary alignment across the evaluated test splits.

Table 2: Localization under stricter IoU thresholds on five test splits. Split entries report Acc@0.7/Acc@0.9; aggregate metrics are computed jointly across the splits. † denotes Hi-R1 aggregates reconstructed from its reported size groups; ‡ marks locally reproduced baselines (Appendix A.6). Bold and underline denote the best and second-best results.

	RefCOCO		RefCOCO+		RefCOCOg	Aggregate Results		
Method	testA	testB	testA	testB	test	mIoU ↑	Acc@0.7 ↑	Acc@0.9 ↑
Hi-R1† (Zhu et al., 2026b)	-	-	-	-	-	79.97	80.07	54.84
InternVL3.5-8B‡ (Wang et al., 2025)	85.9/63.0	77.5/55.0	83.9/61.6	70.6/50.0	79.3/58.6	79.33	79.68	58.02
Qwen3-VL-4B (Bai et al., 2025a; Zhu et al., 2026c)	88.6/–	81.3/–	86.0/–	73.3/–	82.2/–	81.74	82.51	61.24
Qwen3-VL-8B‡ (Bai et al., 2025a)	88.8/66.3	81.1/58.4	86.4/64.9	74.8/54.0	83.0/61.7	82.39	83.09	61.35
Qwen3.5-9B‡ (Qwen Team, 2026)	88.7/65.9	81.7/58.6	86.4/64.5	75.6/54.3	83.1/59.6	82.16	83.31	60.64
CogVLM-Grounding-17B‡ (Wang et al., 2024b)	<u>90.6/69.2</u>	<u>83.0/61.3</u>	<u>88.6/67.9</u>	<u>77.3/57.7</u>	<u>85.5/64.7</u>	<u>83.60</u>	<u>85.31</u>	<u>64.45</u>
CoEvolve (ours)	**92.7/74.2**	**87.1/67.8**	**90.7/73.1**	**82.3/64.7**	**86.9/65.2**	**85.72**	**87.94**	**68.64**

4.4 Controlled Box Correction

Grounding self-correction has been studied through iterative prompting and visual verification (Liao et al., 2025). We evaluate a trained coordinate editor on 30,969 examples using the same deterministic corrupted anchors for all methods. The anchors span multiple IoU levels and geometric perturbations and have 58.80% mIoU; Appendix C.4 gives the full construction protocol. This setting bypasses RER and directly evaluates end-to-end box correction under a shared corrupted-anchor protocol. BDR performs one same-position refinement pass, while the compared baselines generate corrected boxes from the same image, referring expression, and input anchor. Exact state templates, prompts, and parsing rules appear in Appendix A.6.

We report output localization metrics and $\mathrm{Repair@}t$, the fraction of initially sub-threshold anchors corrected to at least t:

$$\mathrm{Repair@}t = \frac{|\{i : u_i < t,\ \hat{u}_i \geq t\}|}{|\{i : u_i < t\}|}, \tag{12}$$

where u_i and $\hat{u}_i$ denote input and output IoUs. We additionally report the mean IoU change, $N^{-1}\sum_i(\hat{u}_i - u_i)$, and the improvement and degradation rates, defined as the fractions of anchors whose output IoU is strictly higher and lower than the corresponding input IoU, respectively.

Fig. 3 shows that BDR raises mIoU to 86.19% and achieves 87.99% $\mathrm{Repair@}0.7$. Among the compared correction baselines, Qwen3.5-9B is strongest in output mIoU. CoEvolve BDR improves output mIoU, $\mathrm{Acc@}0.9$, and $\mathrm{Repair@}0.7$ over this baseline by 2.34, 5.46, and 8.11 points, respectively. The degradation rate is 11.42% for Qwen3.5-9B and 15.50% for CoEvolve BDR.

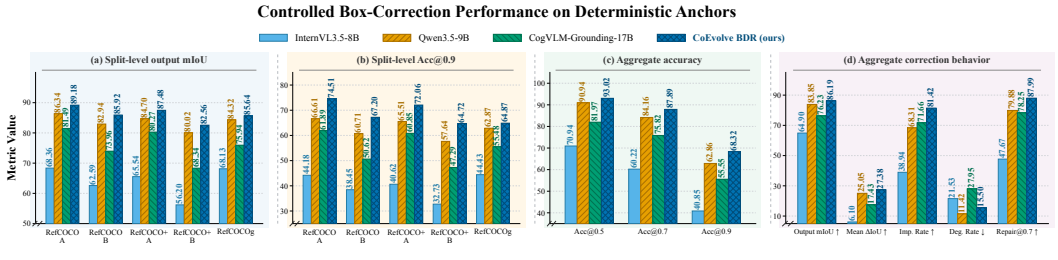

Figure 3: Controlled box correction on five test splits with 30,969 deterministic anchors. Aggregate metrics are computed jointly over all anchors. CoEvolve BDR achieves the highest output mIoU, aggregate localization accuracies, mean IoU gain, improvement rate, and $\mathrm{Repair@}0.7$, while Qwen3.5-9B yields the lowest degradation rate. Higher values are better except for degradation rate.

4.5 State Construction and Source-Matched Refinement

Meaningful-state ablation. Zero-Anchor BDR removes both the upstream localization hypothesis and the anchor-aware preservation and preference objectives enabled by that hypothesis, reducing training to direct same-position coordinate prediction. CoEvolve instead uses the RER state as both a localization starting point and an edit reference. Fig. 4 shows a 2.22-point mIoU gain, while the improvement grows from 2.19 points at $\mathrm{Acc@}0.5$ to 3.68 at $\mathrm{Acc@}0.7$ and 6.48 at $\mathrm{Acc@}0.9$. This widening gap supports the joint value of a meaningful prior localization state and the anchor-aware objectives it enables. Appendix Table 10 reports split-level results.

Structured state construction and source-matched refinement. Under the same structured interface, RER raises mIoU from 63.48% to 80.70% and $\mathrm{Acc@}0.9$ from 24.28 to 59.81 relative to unadapted Qwen3.5-9B. Because both policies use the same response format, these gains show the benefit of RER training for structured region evolution. For each state source, we train a separate BDR from scratch on its outputs. Table 3 shows that native outputs start at 82.16% mIoU, above RER states at 80.70%. After source-matched refinement, the RER pipeline gains 5.02 points and reaches 85.72%, while the native-output pipeline gains 2.41 points and reaches 84.57%. At $\mathrm{Acc@}0.9$, their respective gains are 8.83 and 5.68 points, yielding 68.64% and 66.32%. This reversal shows that initial localization accuracy alone does not determine how effectively a state can be refined.

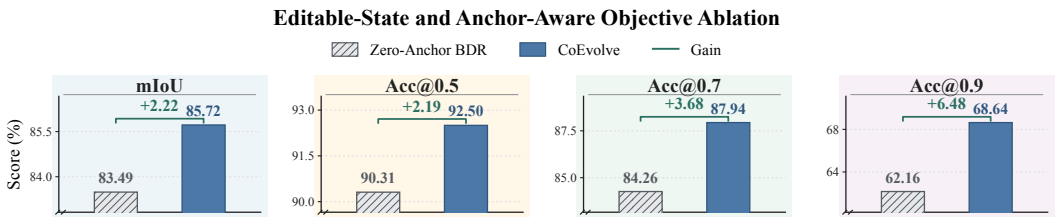

Figure 4: Effect of a meaningful editable state and its associated anchor-aware objectives on five test splits. Aggregate metrics are computed jointly across the splits. Zero-Anchor BDR (ours) uses zero anchors; CoEvolve (ours) uses RER states and anchor-aware objectives. Gains are computed from unrounded aggregate values before display rounding.

Table 3: Source-matched state construction and refinement on five test splits. Each source is paired with a separately trained BDR; metrics are computed jointly across the splits.

State / editor	RER Trained?	Init. mIoU	Final mIoU	Δ mIoU	Init. Acc@0.9	Final Acc@0.9	Δ Acc@0.9
Structured Qwen3.5	No	63.48	–	–	24.28	–	–
Native output → BDR (ours)	No	**82.16**	84.57	+2.41	**60.64**	66.32	+5.68
RER → BDR (ours)	Yes	80.70	**85.72**	**+5.02**	59.81	**68.64**	**+8.83**

4.6 Cross-Domain Adaptation and Downstream Segmentation

Remote-sensing grounding. After domain-specific retraining on DIOR-RSVG, CoEvolve achieves 86.20% Pr@0.5 and 76.23% mIoU. Relative to the reproduced native Qwen3.5-9B baseline, which is evaluated without DIOR-RSVG adaptation, the complete domain-adapted system improves these metrics by 30.35 and 24.40 points, respectively. Table 4 compares CoEvolve with task-specific methods and general-purpose MLLM baselines.

Table 4: Remote-sensing grounding on DIOR-RSVG. Public methods follow their reported protocols; RER and BDR are retrained on the DIOR-RSVG training split. † denotes our reproduced native baseline evaluated without DIOR-RSVG adaptation. Best and second-best results are bolded and underlined, respectively; dashes denote unreported results.

Method	Pr@0.5 ↑	Pr@0.7 ↑	Pr@0.9 ↑	mIoU ↑
LQVG (Lan et al., 2024)	83.41	75.91	**43.53**	74.02
LPVA (Li et al., 2024)	82.27	72.25	39.55	72.35
Qwen3.5-4B (Qwen Team, 2026; Wang et al., 2026a)	53.10	39.40	–	49.00
InternVL3.5-8B (Wang et al., 2025; 2026a)	64.90	50.50	–	56.90
RSGround-R1 (Huang et al., 2026a)	71.80	58.70	–	63.40
GeoSearcher (Wang et al., 2026a)	83.20	72.40	–	73.50
Qwen3.5-9B† (Qwen Team, 2026)	55.85	43.15	18.81	51.83
CoEvolve (ours)	**86.20**	**76.16**	43.45	**76.23**

Downstream segmentation. CoEvolve predictions also serve as box prompts to frozen SAM3 (Carion et al., 2026), yielding 78.19 Macro-8 cIoU across the eight RefCOCO-family splits without mask fine-tuning. Appendix C.6 provides the full split-wise comparison.

5 Conclusion

CoEvolve reformulates visual grounding as a construct-to-edit process: RER externalizes measurable region commitments, and BDR refines their coordinates through bidirectional same-position reconstruction with geometry-level and behavior-level supervision. Experiments across natural images, controlled perturbations, and remote-sensing scenes support this decomposition. The state-source comparison further shows that refinement depends not only on initial localization quality, but also on how the editable state is constructed and how its editor is trained.

AI USE STATEMENT

We used generative AI tools for language polishing throughout the manuscript and to assist with drafting selected passages, including Section 2 (Preliminaries). AI assistance also supported literature searches, reference formatting, and manuscript organization. All AI-assisted content was manually reviewed by the authors. The authors take full responsibility for the final manuscript, including its technical claims, references, and reported results.

ETHICS STATEMENT

Our experiments use publicly available visual grounding datasets and involve neither human-subject experiments nor the collection of new personal data. We have identified no additional ethical concerns specific to the experimental procedures reported in this work.

REPRODUCIBILITY STATEMENT

The Methodology and Experiments sections describe the model design, training objectives, data splits, and evaluation protocols. We have organized the model weights, source code, and experimental configurations, including the random seeds used in our experiments. We will provide the code in the supplementary materials. Following completion of the peer-review process, we will release the model weights, the code and configurations to facilitate independent reproduction of the reported results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. doi: 10.48550/arXiv.2511.21631. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts. In *Int. Conf. Learn. Represent. (ICLR)*, 2026. URL https://openreview.net/forum?id=r35clVtGzw.

Yong Xien Chng, Henry Zheng, Yizeng Han, et al. Mask grounding for referring image segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 26573–26583, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chng_Mask_Grounding_for_Referring_Image_Segmentation_CVPR_2024_paper.html.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, et al. TransVG: End-to-end visual grounding with transformers. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 1769–1779, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Deng_TransVG_End-to-End_Visual_Grounding_With_Transformers_ICCV_2021_paper.html.

Yue Fan, Xuehai He, Diji Yang, et al. GRIT: Teaching MLLMs to think with images. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-3884. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/a92ee7e49a25412add0883454da6bed9-Abstract-Conference.html.

Shiqi Huang, Shuting He, and Bihan Wen. RSGround-R1: Rethinking remote sensing visual grounding through spatial reasoning. *arXiv preprint arXiv:2601.21634*, 2026a. doi: 10.48550/arXiv.2601.21634. URL https://arxiv.org/abs/2601.21634.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, et al. Vision-R1: Incentivizing reasoning capability in multimodal large language models. In *Int. Conf. Learn. Represent. (ICLR)*, 2026b. URL https://openreview.net/forum?id=UZIjskfbfU.

Aishwarya Kamath, Mannat Singh, Yann LeCun, et al. MDETR: Modulated detection for end-to-end multi-modal understanding. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 1780–1790, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Kamath_MDETR_-_Modulated_Detection_for_End-to-End_Multi-Modal_Understanding_ICCV_2021_paper.html.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, et al. ReferItGame: Referring to objects in photographs of natural scenes. In *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, pp. 787–798, 2014. doi: 10.3115/v1/D14-1086. URL https://doi.org/10.3115/v1/D14-1086.

Xin Lai, Zhuotao Tian, Yukang Chen, et al. LISA: Reasoning segmentation via large language model. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9579–9589, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Lai_LISA_Reasoning_Segmentation_via_Large_Language_Model_CVPR_2024_paper.html.

Meng Lan, Fu Rong, Hongzan Jiao, et al. Language query based transformer with multi-scale cross-modal alignment for visual grounding on remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 62:1–14, 2024. doi: 10.1109/TGRS.2024.3407598. URL https://doi.org/10.1109/TGRS.2024.3407598.

Ke Li, Di Wang, Haojie Xu, et al. Language-guided progressive attention for visual grounding in remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 62:1–13, 2024. doi: 10.1109/TGRS.2024.3423663. URL https://doi.org/10.1109/TGRS.2024.3423663.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, et al. Grounded language-image pre-training. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10965–10975, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Li_Grounded_Language-Image_Pre-Training_CVPR_2022_paper.html.

Shufan Li, Jiuxiang Gu, Kangning Liu, et al. Lavida-O: Elastic large masked diffusion models for unified multimodal understanding and generation. In *Int. Conf. Learn. Represent. (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/52d7e1f59be8a140bd5c3b0b5b8bdee2-Abstract-Conference.html.

Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, et al. Can large vision-language models correct semantic grounding errors by themselves? In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 14667–14678, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liao_Can_Large_Vision-Language_Models_Correct_Semantic_Grounding_Errors_By_Themselves_CVPR_2025_paper.html.

Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 23592–23601, 2023a. URL https://openaccess.thecvf.com/content/CVPR2023/html/Liu_GRES_Generalized_Referring_Expression_Segmentation_CVPR_2023_paper.html.

Daqing Liu, Hanwang Zhang, Feng Wu, et al. Learning to assemble neural module tree networks for visual grounding. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 4673–4682, 2019a. doi: 10.1109/ICCV.2019.00477. URL https://doi.org/10.1109/ICCV.2019.00477.

Jiang Liu, Hui Ding, Zhaowei Cai, et al. PolyFormer: Referring image segmentation as sequential polygon generation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 18653–18663, 2023b. URL https://openaccess.thecvf.com/content/CVPR2023/html/Liu_PolyFormer_Referring_Image_Segmentation_As_Sequential_Polygon_Generation_CVPR_2023_paper.html.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 38–55, 2024. doi: 10.1007/978-3-031-72970-6_3. URL https://doi.org/10.1007/978-3-031-72970-6_3.

Xihui Liu, Zihao Wang, Jing Shao, et al. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1950–1959, 2019b. doi: 10.1109/CVPR.2019.00205. URL https://doi.org/10.1109/CVPR.2019.00205.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, et al. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. doi: 10.48550/arXiv.2503.06520. URL https://arxiv.org/abs/2503.06520.

Junhua Mao, Jonathan Huang, Alexander Toshev, et al. Generation and comprehension of unambiguous object descriptions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11–20, 2016. doi: 10.1109/CVPR.2016.9. URL https://doi.org/10.1109/CVPR.2016.9.

Jingyang Ou, Shen Nie, Kaiwen Xue, et al. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *Int. Conf. Learn. Represent. (ICLR)*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Qwen Team. Qwen3.5. Qwen Blog, 2026. URL https://qwen.ai/blog?id=qwen3.5. Accessed: 2026-09-17.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, et al. Simple and effective masked diffusion language models. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Dongwei Sun, Bowen Yao, Yujie Zhang, et al. EchoChange: A diffusion language model with dual pass remasking for factual remote sensing disaster change captioning. *arXiv preprint arXiv:2608.01856*, 2026. doi: 10.48550/arXiv.2608.01856. URL https://arxiv.org/abs/2608.01856.

Mingjie Sun, Jimin Xiao, and Eng Gee Lim. Iterative shrinking for referring expression grounding using deep reinforcement learning. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 14060–14069, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Sun_Iterative_Shrinking_for_Referring_Expression_Grounding_Using_Deep_Reinforcement_Learning_CVPR_2021_paper.html.

Jiajin Tang, Ge Zheng, Cheng Shi, et al. Contrastive grouping with transformer for referring image segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 23570–23580, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Tang_Contrastive_Grouping_With_Transformer_for_Referring_Image_Segmentation_CVPR_2023_paper.html.

Dianyu Wang, Peirong Zhang, Xuyang Li, et al. GeoSearcher: Anchor-guided progressive reasoning for remote sensing visual grounding with process supervision. *arXiv preprint arXiv:2607.01050*, 2026a. doi: 10.48550/arXiv.2607.01050. URL https://arxiv.org/abs/2607.01050.

Hao Wang, Limeng Qiao, Zequn Jie, et al. X-SAM: From segment anything to any segmentation. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, volume 40, pp. 26187–26196, 2026b. doi: 10.1609/aaai.v40i31.39822. URL https://doi.org/10.1609/aaai.v40i31.39822.

Peng Wang, Shijie Wang, Junyang Lin, et al. ONE-PEACE: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023. doi: 10.48550/arXiv.2305.11172. URL https://arxiv.org/abs/2305.11172.

Peng Wang, Shuai Bai, Sinan Tan, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a. doi: 10.48550/arXiv.2409.12191. URL https://arxiv.org/abs/2409.12191.

Weihan Wang, Qingsong Lv, Wenmeng Yu, et al. CogVLM: Visual expert for pretrained language models. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 37, 2024b. doi: 10.52202/079017-3860. URL https://doi.org/10.52202/079017-3860.

Weiyun Wang, Zhangwei Gao, Lixin Gu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. doi: 10.48550/arXiv.2508.18265. URL https://arxiv.org/abs/2508.18265.

Zhaoqing Wang, Yu Lu, Qiang Li, et al. CRIS: CLIP-driven referring image segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11686–11695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_CRIS_CLIP-Driven_Referring_Image_Segmentation_CVPR_2022_paper.html.

Cong Wei, Yujie Zhong, Haoxian Tan, et al. HyperSeg: Hybrid segmentation assistant with fine-grained visual perceiver. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 8931–8941, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wei_HyperSeg_Hybrid_Segmentation_Assistant_with_Fine-grained_Visual_Perceiver_CVPR_2025_paper.html.

Cong Wei, Yujie Zhong, Haoxian Tan, et al. InstructSeg: Unifying instructed visual segmentation with multi-modal large language models. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 20193–20203, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wei_InstructSeg_Unifying_Instructed_Visual_Segmentation_with_Multi-modal_Large_Language_Models_ICCV_2025_paper.html.

Qiong Wu, Xiangcong Yang, Yiyi Zhou, et al. Grounded chain-of-thought for multimodal large language models. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 33577–33587, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wu_Grounded_Chain-of-Thought_for_Multimodal_Large_Language_Models_CVPR_2026_paper.html.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, et al. DeepSeek-VL2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024. doi: 10.48550/arXiv.2412.10302. URL https://arxiv.org/abs/2412.10302.

Linhui Xiao, Xiaoshan Yang, Fang Peng, et al. OneRef: Unified one-tower expression grounding and segmentation with mask referring modeling. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-4438. URL https://doi.org/10.52202/079017-4438.

Bin Yan, Yi Jiang, Jiannan Wu, et al. Universal instance perception as object discovery and retrieval. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 15325–15336, 2023. doi: 10.1109/CVPR52729.2023.01471. URL https://doi.org/10.1109/CVPR52729.2023.01471.

Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 4644–4653, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Yang_Dynamic_Graph_Attention_for_Referring_Expression_Comprehension_ICCV_2019_paper.html.

Zhao Yang, Jiaqi Wang, Yansong Tang, et al. LAVT: Language-aware vision transformer for referring image segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 18155–18165, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Yang_LAVT_Language-Aware_Vision_Transformer_for_Referring_Image_Segmentation_CVPR_2022_paper.html.

Haoxuan You, Haotian Zhang, Zhe Gan, et al. Ferret: Refer and ground anything anywhere at any granularity. In *Int. Conf. Learn. Represent. (ICLR)*, 2024. URL https://openreview.net/forum?id=2msbbX3ydD.

Licheng Yu, Patrick Poirson, Shan Yang, et al. Modeling context in referring expressions. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 69–85, 2016. doi: 10.1007/978-3-319-46475-6_5. URL https://doi.org/10.1007/978-3-319-46475-6_5.

Licheng Yu, Zhe Lin, Xiaohui Shen, et al. MAttNet: Modular attention network for referring expression comprehension. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1307–1315, 2018. doi: 10.1109/CVPR.2018.00142. URL https://doi.org/10.1109/CVPR.2018.00142.

Guanqi Zhan, Changye Li, Zhijian Liu, et al. Scaling test-time inference for visual grounding. *arXiv preprint arXiv:2601.13633*, 2026. doi: 10.48550/arXiv.2601.13633. URL https://arxiv.org/abs/2601.13633.

Yang Zhan, Zhitong Xiong, and Yuan Yuan. RSVG: Exploring data and models for visual grounding on remote sensing data. *IEEE Trans. Geosci. Remote Sens.*, 61:1–13, 2023. doi: 10.1109/TGRS.2023.3250471. URL https://doi.org/10.1109/TGRS.2023.3250471.

Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4158–4166, 2018. doi: 10.1109/CVPR.2018.00437. URL https://doi.org/10.1109/CVPR.2018.00437.

Haotian Zhang, Haoxuan You, Philipp Dufter, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. In *Conf. Lang. Model. (COLM)*, 2024. URL https://openreview.net/forum?id=EEPBOB2Xww.

Yuxuan Zhang, Tianheng Cheng, Rui Hu, et al. EVF-SAM: Early vision-language fusion for text-prompted segment anything model. *Image Vis. Comput.*, 173:106028, 2026a. doi: 10.1016/j.imavis.2026.106028. URL https://doi.org/10.1016/j.imavis.2026.106028.

Zekang Zhang, Guangyu Gao, Youyun Tang, et al. FlowSeg: Dynamic semantic guidance for LLM-conditioned segmentation. In *Int. Conf. Mach. Learn. (ICML)*, 2026b. URL https://icml.cc/virtual/2026/poster/63754.

Jinguo Zhu, Weiyun Wang, Zhe Chen, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. doi: 10.48550/arXiv.2504.10479. URL https://arxiv.org/abs/2504.10479.

Lianghui Zhu, Bin Ouyang, Yuxuan Zhang, et al. LENS: Learning to segment anything with unified reinforced reasoning. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, volume 40, pp. 13952–13960, 2026a. doi: 10.1609/aaai.v40i16.38405. URL https://doi.org/10.1609/aaai.v40i16.38405.

Xiuyuan Zhu, Ke Lu, Kun Dong, et al. Hi-Token: Hierarchical coordinate tokenization for generative visual grounding. *arXiv preprint arXiv:2608.03471*, 2026b. doi: 10.48550/arXiv.2608.03471. URL https://arxiv.org/abs/2608.03471.

Xiuyuan Zhu, Ke Lu, Hao Wu, et al. IoU-PD: IoU-aware privileged distillation for visual grounding with multimodal large language models. *arXiv preprint arXiv:2607.15732*, 2026c. doi: 10.48550/arXiv.2607.15732. URL https://arxiv.org/abs/2607.15732.

APPENDIX CONTENTS

A Supplementary Method Details 16
A.1 RER–BDR Interface 16
A.2 Sequential Training and BDR Data 16
A.3 Region-Evolution Reinforcement 16
A.4 BDR Refinement and Targets 18
A.5 Inference Procedure 21
A.6 Prompts and Evaluation 21
A.7 Implementation Notes 23

B Formal Properties 23

C Additional Experimental Results 24
C.1 Dataset Accounting 24
C.2 RER Localization State Quality 24
C.3 Qualitative Grounding Results 26
C.4 Controlled Correction Details 28
C.5 Remote-Sensing Grounding 29
C.6 Downstream Segmentation 30

D Limitations 30

A SUPPLEMENTARY METHOD DETAILS

This appendix gives the complete mathematical specification of RER and BDR. We use the same notation as in the main paper and make explicit which quantities are used only to construct training targets and which quantities are visible to the model.

A.1 NOTATION AND THE RER–BDR INTERFACE

For an image–expression pair (I, q), a box is represented in the normalized integer coordinate system by $\mathbf{b} = [x_1, y_1, x_2, y_2] \in [0, 1000]^4$, with $x_1 < x_2$ and $y_1 < y_2$. The annotated box is $\mathbf{b}^*$. RER produces a response

$$y = \big\langle \texttt{<refine>}\ (r_1, \mathbf{b}_1), \ldots, (r_K, \mathbf{b}_K)\ \texttt{</refine>}\ \texttt{<answer>}\ \mathbf{b}_{\text{ans}}\ \texttt{</answer>} \big\rangle, \tag{13}$$

where $K \in \{3, \ldots, 5\}$ is the number of refinement steps. BDR receives this complete response, including its text and all coordinate tokens, as an editable localization state. RER and BDR are optimized sequentially and communicate only through this generated state.

A.2 SEQUENTIAL TRAINING AND BDR DATASET CONSTRUCTION

CoEvolve is trained sequentially rather than end to end. We first optimize RER on the RefCOCOg training set using the rewards defined below. After RER training is complete, its adapter is frozen and used to generate one complete localization response for each RefCOCOg training expression. Each saved record pairs the original image and referring expression with the complete RER assistant response and its annotated box. The generated RER responses are then used to construct the BDR training set.

BDR initializes a separate adapter on the frozen Qwen3.5-9B backbone; it is not initialized from the RER adapter. At loading time, auxiliary <think> content is removed, every coordinate is zero-padded to four digits, and the remaining <refine> and <answer> sequence is used as the editable input state. The unmodified records retain the natural errors produced by RER. Eligible high-overlap states are additionally augmented with synchronized geometric perturbations, as detailed in Section A.4, so BDR observes both the natural RER error distribution and deliberately corrupted anchors. The annotation is retained only for target construction and loss computation and is never inserted into the model input.

A.3 REGION-EVOLUTION REINFORCEMENT

Interface adaptation and state construction. RER adapts the backbone from terminal box generation to a structured interface in which each reasoning Step exposes an explicit region. The terminal annotation evaluates both the final answer and properties of the generated trajectory, deriving trajectory-level spatial rewards from terminal annotations without introducing manually labeled rationales or intermediate boxes. The resulting response is therefore optimized as a localization state for subsequent evaluation and editing, rather than as explanation text alone.

Structural rewards. For a non-empty target, let n be the number of Step headers. The format reward is

$$R_{\text{fmt}} = \begin{cases} 0, & \text{invalid outer tagged structure,} \\ 0.5, & \text{valid outer structure and } (n < 3 \text{ or } n > 5), \\ 1, & \text{valid outer structure and } 3 \le n \le 5. \end{cases} \tag{14}$$

All evaluated benchmarks use non-empty targets; the empty-target branch is retained as a generic interface fallback. For an empty target, a valid tagged response with an empty answer receives one, without a minimum Step count. The step-format reward checks every Step independently and is

$$R_{\text{step}} = \begin{cases} R_{\text{fmt}}, & \text{empty target,} \\ R_{\text{fmt}}\, \mathbf{1}[\text{each Step has a parseable final bracketed box}], & \text{otherwise.} \end{cases} \tag{15}$$

When a Step contains several bracketed expressions, the last one is used as the Step box. Thus, the reward evaluates the final state written by a Step rather than an earlier box mentioned in its explanation.

Terminal box reward. For a valid single predicted box, let $u = \mathrm{IoU}(\mathbf{b}_{\mathrm{ans}}, \mathbf{b}^*)$ and $g = \mathrm{GIoU}(\mathbf{b}_{\mathrm{ans}}, \mathbf{b}^*)$. The generalized overlap is computed from the minimum enclosing rectangle C and its area A_C:

$$\mathrm{GIoU} = u - \frac{A_C - A_\cup}{A_C}, \tag{16}$$

where $A_\cup$ is the union area. The scale and aspect agreement term is

$$S_A = \frac{1}{2}\left(\frac{\min(A_{\mathrm{ans}}, A^*)}{\max(A_{\mathrm{ans}}, A^*)} + \frac{\min(\rho_{\mathrm{ans}}, \rho^*)}{\max(\rho_{\mathrm{ans}}, \rho^*)}\right), \qquad \rho = \frac{\text{width}}{\text{height}}. \tag{17}$$

The raw terminal score and its format-modulated form are

$$R_{\mathrm{box}}^{\mathrm{raw}} = u + 0.5u^2 + 0.2g + 0.2\,\sigma(10(u-0.5))\,S_A, \tag{18}$$

$$R_{\mathrm{box}} = R_{\mathrm{fmt}}\max(0, R_{\mathrm{box}}^{\mathrm{raw}}) + \min(0, R_{\mathrm{box}}^{\mathrm{raw}}). \tag{19}$$

Here $\sigma(\cdot)$ denotes the logistic sigmoid. The implementation assigns -0.5 to a malformed or missing answer, an answer with the wrong number of boxes, or an answer containing multiple boxes. A correct empty-target answer receives one. Negative geometry feedback is never attenuated by an invalid format score.

Progressive reward. Let $u_i = \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}^*)$. To provide a distance signal before overlap is nonzero, the center quality of Step i is

$$q_i = 1 - \frac{d_i^{\mathrm{norm}}}{\sqrt{2}}, \tag{20}$$

where d_i^{norm} is the Euclidean center distance after mapping coordinates to $[0,1]^2$. Let d_i^{center} be the Euclidean distance between the centers of $\mathbf{b}_i$ and $\mathbf{b}^*$ in the original $[0,1000]^2$ coordinate system. The first Step receives a coverage score

$$C_{\mathrm{cov}} = \frac{1}{2}\left(C_{\mathrm{area}} + \exp\left[-d_1^{\mathrm{center}}/350\right]\right). \tag{21}$$

Let A_1 and A^* denote the first-Step and target areas, and set $A_{\mathrm{req}} = \max(2A^*, 10^4)$. The implemented area term is

$$C_{\mathrm{area}} = \begin{cases} 1, & A_1 \geq 5 \times 10^5, \\ \sigma(6(A_1/A_{\mathrm{req}} - 1)), & \text{otherwise.} \end{cases} \tag{22}$$

For later Steps, RER compares each box with the best preceding state. We use $\epsilon = 10^{-8}$ to stabilize the remaining-improvement denominators:

$$\Delta_i^u = \frac{u_i - \max_{j<i} u_j}{1 - u_1 + \epsilon}, \qquad \Delta_i^q = \frac{q_i - \max_{j<i} q_j}{1 - q_1 + \epsilon},$$

$$g_i = 0.6\Delta_i^u + 0.4\Delta_i^q, \qquad C_{\mathrm{prog}} = \max\left(0, \sum_{i=2}^{K} g_i\right). \tag{23}$$

The overall first-to-last gain and the last-Step/answer consistency are

$$C_{\mathrm{overall}} = \max\left(0, \frac{u_K - u_1}{1 - u_1 + \epsilon}\right), \qquad C_{\mathrm{cons}} = \mathrm{IoU}(\mathbf{b}_K, \mathbf{b}_{\mathrm{ans}}). \tag{24}$$

The progressive reward and the complete RER reward are

$$R_{\mathrm{prog}} = R_{\mathrm{fmt}}\left(0.20C_{\mathrm{cov}} + 0.50C_{\mathrm{prog}} + 0.20C_{\mathrm{overall}} + 0.10C_{\mathrm{cons}}\right), \tag{25}$$

$$R_{\mathrm{RER}} = 0.5R_{\mathrm{fmt}} + 0.5R_{\mathrm{step}} + R_{\mathrm{prog}} + R_{\mathrm{box}}. \tag{26}$$

If the trajectory has fewer than three or more than five Steps, or any Step does not contain a valid positive-area box, R_{prog} is zero. In implementation, Eq. (26) is realized by supplying $(R_{\mathrm{fmt}}, R_{\mathrm{step}}, R_{\mathrm{prog}}, R_{\mathrm{box}})$ to GRPO with weights $0.5, 0.5, 1.0, 1.0$, respectively; these weights are not applied again to an already combined reward. For each prompt, eight sampled responses form one GRPO group. The annotation is used to evaluate geometry, but no human-written reasoning chain is used as a token target.

Figure 5 places the corresponding training diagnostics beside the reward specification. The curves show the total reward together with the structural, progressive, and terminal geometry components.

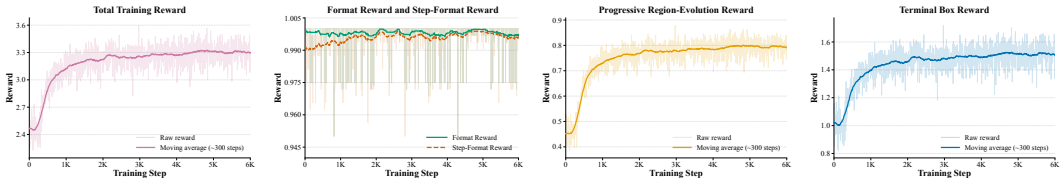

Figure 5: Supplementary training dynamics for RER. For visualization, we show the first 6K optimization steps; this window does not indicate the total training duration. From left to right: total reward; format and Step-format rewards; progressive region-evolution reward; and terminal box reward.

A.4 Bidirectional Same-Position Refinement and Target Construction

Editable state and masks. Every coordinate is written as four zero-padded digit tokens, so one box contains sixteen digit positions. Let $\mathcal{M}_{\text{all}}$ be all box digit positions in the assistant response, $\mathcal{M}_b$ the positions of the last refinement box and the answer, and $\mathcal{M}_f$ the structural tags and punctuation. The normal strongly supervised region therefore has $|\mathcal{M}_b| = 32$.

BDR removes causal masking from the base model's Full Attention layers, providing each same-position prediction with bidirectional access to the global response context. If $\mathbf{z} = F_\phi(I, q, \mathbf{x}^{\text{in}})$ are logits aligned with the observed response, a digit candidate is

$$\widehat{x}_t = \arg\max_{v \in \mathcal{V}_{\text{digit}}} z_{t,v}, \qquad t \in \mathcal{M}_{\text{all}}, \tag{27}$$

where $\mathcal{V}_{\text{digit}}$ contains the ten single-character digits. The 16 candidate digits associated with each box role are decoded separately into one candidate box.

Same-position reconstruction predicts every editable coordinate from the same observed response and retains alignment with its original response position. Keeping the observed coordinate tokens visible provides an explicit input reference against which candidate edits can be evaluated and accurate coordinates can be preserved. The decoded box therefore provides the synchronized candidate used by coordinate regression, quality-gated target selection, entropy regularization, and relative preference between the input and output.

Behavior-Level Supervision: Perturbation and Quality-Gated Targets. When the RER answer has IoU above 0.5, it is selected for perturbation with probability 0.3. The selected answer is replaced by a nonzero box obtained through rejection sampling toward the configured IoU interval $[0.2, 0.5]$ relative to the annotated box. The sampler uses translation, shrink, expansion, or random geometry and rejects zero-area boxes; after 200 unsuccessful attempts it uses bounded coordinate offsets. The answer, the last refinement box, and earlier Steps whose IoU with the original answer exceeds 0.95 are replaced together to preserve spatial consistency among the perturbed coordinate fields.

At each rejection-sampling attempt, the four proposal families are selected with equal probability. Their exact ranges are summarized in Table 5. Every proposal is clipped to the normalized coordinate range and accepted only when it has positive area and $\text{IoU}(\mathbf{b}_{\text{proposal}}, \mathbf{b}^*) \in [0.2, 0.5]$. The fallback independently offsets the two horizontal coordinates within ± 0.8 target widths and the two vertical coordinates within ± 0.8 target heights before clipping.

In the coordinate-shift proposal, w and h denote the width and height of the ground-truth box, and s is the sampled offset-scale factor.

Let $\mathbf{b}^{\text{in}}_{\text{ans}}$ denote the actual Answer box seen by BDR after this operation, and let $\widetilde{\mathbf{b}}_{\text{ans}}$ be the hard candidate decoded from its output. With $\eta = 0.9$ and $\gamma = 0.005$, the dynamic-copy condition is

$$\mathcal{C}_{\text{out}} \equiv \Big(\text{IoU}(\widetilde{\mathbf{b}}_{\text{ans}}, \mathbf{b}^*) > \eta\Big) \wedge \Big(\text{IoU}(\widetilde{\mathbf{b}}_{\text{ans}}, \mathbf{b}^*) \geq \text{IoU}(\mathbf{b}^{\text{in}}_{\text{ans}}, \mathbf{b}^*) + \gamma\Big). \tag{28}$$

Table 5: Geometric proposals used to construct BDR training perturbations. Probabilities refer to proposal selection at each rejection-sampling attempt; the accepted proportions may differ because proposals are filtered by the target IoU interval $[0.2, 0.5]$.

Proposal	Probability	Parameter range
Coordinate shift	0.25	$s \sim \mathcal{U}(0.2, 1.5)$; offsets in $[-sw, sw]$ and $[-sh, sh]$
Centered shrink	0.25	width and height ratio $r \sim \mathcal{U}(0.3, 0.9)$
Centered expansion	0.25	width and height ratio $r \sim \mathcal{U}(1.3, 3.0)$
Random box	0.25	valid integer endpoints in $[0, 1000]$

If it is not activated and the unperturbed input has IoU at least η, the static preservation branch is used. The selected absolute target is

$$\mathbf{b}^{\mathrm{tar}} = \begin{cases} \widetilde{\mathbf{b}}_{\mathrm{ans}}, & \text{if } \mathcal{C}_{\mathrm{out}}, \\ \mathbf{b}^{\mathrm{in}}_{\mathrm{ans}}, & \text{unperturbed and } \mathrm{IoU}(\mathbf{b}^{\mathrm{in}}_{\mathrm{ans}}, \mathbf{b}^{*}) \geq \eta, \\ \mathbf{b}^{*}, & \text{otherwise.} \end{cases} \tag{29}$$

The ground-truth box is used to construct this training target and the geometry labels only; it is never provided as an input during inference.

Geometry-Level Supervision. For a target digit y_t, cross-entropy over positions $\mathcal{M}$ is

$$\mathcal{L}_{\mathrm{CE}}(\mathcal{M}) = -\frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \log \frac{\exp z_{t,y_t}}{\sum_v \exp z_{t,v}}. \tag{30}$$

Let $\widehat{b}^{\mathrm{hard}}_{sk}$ denote coordinate k of the hard argmax-decoded candidate box for role s. For the final Step and answer, the four digits belonging to coordinate k share one detached weight obtained by averaging the two hard-coordinate errors:

$$w_k = 1 + \frac{w_{\mathrm{L1CE}}}{2} \sum_{s \in \{\mathrm{last}, \mathrm{ans}\}} \frac{\left| \widehat{b}^{\mathrm{hard}}_{sk} - b^{\mathrm{tar}}_{k} \right|}{1000}, \qquad w_{\mathrm{L1CE}} = 10. \tag{31}$$

Each digit position t of coordinate k inherits $w_t = w_k$, and CE_t denotes its token cross-entropy. The weighted BBox CE divides by the sum of token weights:

$$\mathcal{L}_{\mathrm{box}} = \frac{\sum_{t \in \mathcal{M}_b} w_t \, \mathrm{CE}_t}{\sum_{t \in \mathcal{M}_b} w_t}. \tag{32}$$

Consequently, L1-derived weights redistribute the gradient toward inaccurate coordinates without changing the nominal CE scale.

The differentiable coordinate used by the regression term is obtained from the ten-digit softmax at each digit position:

$$\begin{aligned} p_{skj}(d) &= \frac{\exp z_{t_{skj},d}}{\sum_{a=0}^{9} \exp z_{t_{skj},a}}, \\ \bar{b}_{sk} &= \sum_{j=0}^{3} 10^{3-j} \sum_{d=0}^{9} d \, p_{skj}(d), \\ \mathcal{L}_{\mathrm{reg}} &= \frac{1}{8} \sum_{s \in \{\mathrm{last}, \mathrm{ans}\}} \sum_{k=1}^{4} \frac{|\bar{b}_{sk} - b^{\mathrm{tar}}_{k}|}{1000}. \end{aligned} \tag{33}$$

The code computes the same quantity as the mean over the eight supervised coordinates. CE preserves the discrete coordinate representation, whereas the regression term supplies a continuous numerical direction.

State Preservation and Confidence Regularization. General reconstruction is applied to assistant tokens outside the coordinate mask. In the warm-up phase it excludes only $\mathcal{M}_b$, so a perturbed historical Step can still receive the selected target through general CE. After warm-up it excludes $\mathcal{M}_{\text{all}}$ and therefore preserves only non-coordinate assistant content. Format CE is computed on $\mathcal{M}_f$, including structural tags and coordinate punctuation.

For the ten-digit distribution p_t, natural entropy is

$$H(p_t) = -\sum_{d=0}^{9} p_t(d) \log p_t(d). \tag{34}$$

The historical entropy term averages $H(p_t)$ over $\mathcal{M}_{\text{all}} \setminus \mathcal{M}_b$. The final entropy term uses $\mathcal{M}_b$ but keeps only positions whose digit-vocabulary argmax already equals the selected target digit. Thus the former sharpens unsupervised historical digits, while the latter does not make an incorrect final digit more confident.

Coordinate-wise Edit Preference. For a fixed sample i and every box role s (all historical Steps, the last Step, and the answer), let $\mathcal{C}_{sk}$ be the changed digit positions in coordinate k. The input and output coordinate errors are compared to the original annotation:

$$\Delta_{sk} = \frac{|\breve{b}^{\text{in}}_{sk} - b^*_k| - |\widetilde{b}^{\text{out}}_{sk} - b^*_k|}{1000}. \tag{35}$$

Only coordinates with $\mathcal{C}_{sk} \neq \varnothing$ and $|\Delta_{sk}| > m$ with $m = 0.005$ are active; their index set is $\mathcal{A}_s = \{k : \mathcal{C}_{sk} \neq \varnothing,\ |\Delta_{sk}| > m\}$. Let x^{in}_t denote the observed input digit token and $\widetilde{x}^{\text{out}}_t$ the digit-vocabulary argmax at position t. On changed digit positions,

$$D_{sk} = \frac{1}{|\mathcal{C}_{sk}|} \sum_{t \in \mathcal{C}_{sk}} \left(z_{t,\widetilde{x}^{\text{out}}_t} - z_{t,x^{\text{in}}_t} \right), \qquad y_{sk} = \mathbf{1}[\Delta_{sk} > m]. \tag{36}$$

The coordinate loss is the expanded binary cross-entropy with logits:

$$\ell_{sk} = -y_{sk} \log \sigma(D_{sk}) - (1 - y_{sk}) \log \left(1 - \sigma(D_{sk})\right), \qquad \sigma(D) = \frac{1}{1 + \exp(-D)}. \tag{37}$$

For $\mathcal{A}_s \neq \varnothing$, active coordinates within one role are weighted by $a_{sk} = \min(1, |\Delta_{sk}|/0.05)$:

$$\mathcal{L}^{(i)}_{\text{pref},s} = \frac{\sum_{k \in \mathcal{A}_s} a_{sk} \ell_{sk}}{\sum_{k \in \mathcal{A}_s} a_{sk}}. \tag{38}$$

We set $\mathcal{L}^{(i)}_{\text{pref},s} = 0$ when $\mathcal{A}_s = \varnothing$. The answer and the last Step have role weight 0.5, and every earlier Step has role weight 0.2. The per-sample and minibatch losses are

$$\mathcal{L}^{(i)}_{\text{pref}} = \sum_{s \in \mathcal{S}} w_s \mathcal{L}^{(i)}_{\text{pref},s}, \qquad \mathcal{L}_{\text{pref}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}^{(i)}_{\text{pref}}. \tag{39}$$

The preference branches are added directly to the base loss. Hence a sample can contribute one active preference term per role, while coordinates inside each role are normalized by their own active magnitude weights. Samples without an active preference coordinate contribute zero but remain in the minibatch denominator N.

Phase-specific objectives. During the first 3% of optimizer updates, the training loss is

$$\mathcal{L}_{\text{early}} = 1.0\, \mathcal{L}_{\text{box}} + 0.2\, \mathcal{L}_{\text{gen}}. \tag{40}$$

Perturbed historical Step labels are covered by the general CE mask in this phase, and 10% of eligible assistant input tokens are replaced by the configured mask token. No token masking is applied after warm-up. BDR then uses

$$\begin{aligned} \mathcal{L}_{\text{mid}} = {} & 1.0\, \mathcal{L}_{\text{box}} + 30\, \mathcal{L}_{\text{reg}} + 0.5\, \mathcal{L}_{\text{fmt}} + 0.2\, \mathcal{L}_{\text{gen}} \\ & + 0.2\, \mathcal{H}_{\text{hist}} + 0.2\, \mathcal{H}_{\text{fin}} + \mathcal{L}_{\text{pref}}. \end{aligned} \tag{41}$$

Only the last Step and answer enter $\mathcal{L}_{\text{box}}$, $\mathcal{L}_{\text{reg}}$, and $\mathcal{H}_{\text{fin}}$; historical boxes enter $\mathcal{H}_{\text{hist}}$. The total sample losses are divided by the effective minibatch size before back-propagation.

Figure 6 follows the complete BDR objective and reports the optimization dynamics of its coordinate, reconstruction, confidence, and preference terms.

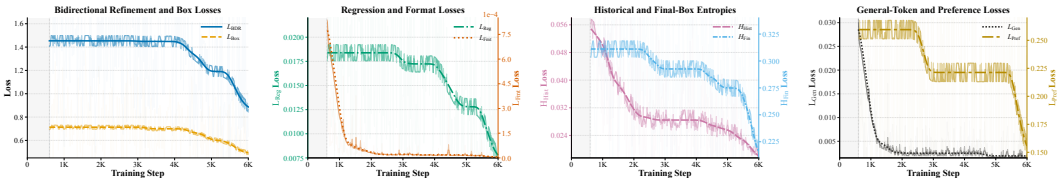

Figure 6: Supplementary training dynamics for BDR. For visualization, we show the first 6K optimization steps; this window does not indicate the total training duration. From left to right: total and coordinate cross-entropy losses; coordinate regression and format losses; historical and final coordinate entropies; and general reconstruction and coordinate-wise preference loss.

A.5 Inference Procedure

At inference, RER first generates the complete localization state. BDR then performs a deterministic same-position pass on the current state. After each pass, only the editable coordinate positions are updated; the surrounding reasoning text and response structure are retained. The response is then detokenized and tokenized again, coordinate masks are rebuilt, and the updated state becomes the next-pass input. We use two BDR passes for all standard natural-image and remote-sensing evaluations, providing a consistent refinement budget across settings. Unparseable final <answer> boxes remain in the evaluation denominator and receive zero IoU.

A.6 Prompt and Evaluation Protocols

Figure 7 consolidates the standard-grounding and controlled-correction interfaces in one prompt plate. The structured base condition and RER use the same structured prompt and parsing rule; the former uses the frozen backbone directly, whereas the latter activates the RER adapter. In controlled correction, every method receives the same image, referring expression q, and deterministic input anchor $\mathbf{b}^{\text{in}}$. The displayed line breaks are typographic, and the placeholders are replaced by the corresponding sample values at inference.

Reproduced standard-grounding baselines. The reproduced entries in Table 2 use each model's native grounding interface rather than the structured RER prompt. Their shared evaluation conventions are summarized in Table 6. Each model produces one autoregressive response per example. Predictions are evaluated as normalized $xyxy$ coordinates in $[0, 1000]$; a single complete final box is extracted, while invalid or unparseable outputs remain in the denominator with zero IoU.

Table 6: Protocols for the locally reproduced standard-grounding baselines in Table 2. "Native" denotes the prompt interface distributed or recommended for the corresponding model family.

Model	Native prompt interface
InternVL3.5-8B	Referring box with <ref> expression
Qwen3-VL-8B	JSON grounding with normalized box
Qwen3.5-9B	JSON grounding with normalized box
CogVLM-Grounding-17B	Caption-to-box grounding

Native-output-derived state construction. For the Native-Output BDR condition, unadapted Qwen3.5-9B first produces visible natural-language analysis and a terminal box through the native grounding interface in Figure 7. Auxiliary <think> content is removed. The remaining visible analysis, including its terminal box, is retained verbatim inside a <refine> block; the same zero-padded box is placed in <answer>. For example, the processed state for the query "the man in yellow coat" is

> **Example native-output-derived state.** <refine> Got it, let's look at the image. The man in the yellow coat is the one on the right. So we need to find his bounding box. Let's check the coordinates. The yellow coat, red pants, and the bib. So the re-

Prompt interfaces used in CoEvolve evaluations

(a) Standard grounding

Native Qwen3.5 interface `<image>`
Locate every object that matches the description "q" in the image. Report bbox coordinates in JSON format using `[x1, y1, x2, y2]` normalized to 0–1000.

Structured Qwen3.5 and RER interface

User prompt You are a precise visual grounding assistant. Locate the target object described in the text. You must reason step by step, progressively narrowing from a broad scene region to a tight box. This **coarse-to-fine** process forces you to deeply understand the whole image and the object's context, preventing misidentification and leading to a more accurate final localization.

Follow the structure and style of the example below.

`<refine>`
Step 1: The scene shows a wooden desk with a laptop, a cup, a blue notebook, and a blue magazine. → [100, 50, 900, 850]
Step 2: Focus on the right side of the desk, ignoring the laptop and the cup on the left; both the blue notebook and the blue magazine are visible in this area. → [400, 200, 850, 750]
Step 3: Among the two blue items, the notebook is lying flat near the desk edge, while the magazine is further back and partially underneath it; identify the notebook as the closer one. → [455, 325, 820, 700]
Step 4: Zoom in on the blue notebook itself, excluding the magazine completely; its cover is facing up with a pen resting beside it. → [510, 345, 750, 650]
Step 5: Tightly enclose only the blue notebook—its full cover and spine—trimming away the pen and all surrounding background. → [523, 382, 719, 620]
`</refine>`
`<answer>`[523, 382, 719, 620]`</answer>`

Requirements Use **3–5 steps**, each with **exactly one bounding box** `[x1, y1, x2, y2]` (integers 0–1000). Each box must be tighter than the previous, covering less background and fitting the target more precisely. For each step, first provide a brief scene observation locating the target; then use → as a delimiter followed by the box. The final `<answer>` box must be **identical to the last step's box**. Output only the two XML tags; no text outside them. If no matching object exists, the last step's box and `<answer>` must both be `[]`.

`<image>`
Text description: q

(b) BDR refinement

User prompt You are a precise visual grounding refinement assistant. Your job is to verify and refine bounding box coordinates through multi-step reasoning by carefully examining the image.

Key rules • If the bounding box in `<answer>` already tightly and accurately encloses the target object → **KEEP IT EXACTLY AS-IS**. Do not change coordinates that are already correct.
• The final `<answer>` is the authoritative result. The last step's bbox in `<refine>` **MUST be consistent** with the `<answer>` bbox.
• If you believe the reasoning chain contains errors, you may modify the relevant steps—update their bounding boxes as needed. You do NOT need to make every single step's bbox perfectly precise; intermediate steps serve as progressive reasoning toward the final answer.

Guidelines • Carefully examine the image to locate the target object described in the text description below.
• If the answer box is already precise → output the SAME reasoning chain unchanged.
• If the reasoning chain has inaccuracies → refine the relevant steps and the answer progressively, with each corrected step moving closer to the correct location.
• Keep the same number of steps and overall structure; adjust coordinates only where they are wrong.
• All coordinates use xyxy format normalized to 0–1000, **zero-padded to 4 digits**. For example: 585 → "0585", 0 → "0000", 1000 → "1000"; a complete bbox is `[0585, 0173, 0797, 0710]`.
• The final `<answer>` box must tightly enclose only the target object, trimming away all surrounding background.

`<image>`
Text description: q

(c) Controlled box correction baselines

Qwen3.5-9B A candidate bounding box $\mathbf{b}^{\text{in}}$ is provided for "q" but **may be inaccurate**. Locate every object that matches the description "q" in the image. Report bbox coordinates in JSON format using `[x1, y1, x2, y2]` normalized to 0–1000.

InternVL3.5-8B `<image>` Please provide the bounding box coordinate of `<ref>`q`</ref>` in this image. A candidate box $[\mathbf{b}^{\text{in}}]$ is provided as reference but **may be inaccurate**.

CogVLM-Grounding-17B The candidate bounding box for q is $[\mathbf{b}^{\text{in}}]$, but it may be inaccurate. Can you **point out** q in the image and provide the **corrected bounding box** of its location?

Figure 7: Exact prompt templates used for standard grounding, BDR refinement, and controlled box correction. Red bold text marks the constraints that define the output interface or the preserve-or-correct behavior. Only line wrapping and sample-specific placeholders are changed for typesetting.

gion is `[0585, 0185, 0793, 0696].</refine><answer>[0585, 0185, 0793, 0696]</answer>`

This state contains a terminal localization hypothesis but no RER-trained ordered region commitments. BDR keeps the `<refine>` content fixed and performs same-position editing only on the 16 Answer digits. A separate native-output editor is trained from scratch on these native-output-derived training states; its parameters are independent of the BDR trained on RER-generated states. Training uses Answer coordinate cross-entropy, soft-coordinate regression, correctness-gated Answer entropy, coordinate-wise preference, format cross-entropy, and general reconstruction of the unchanged `<refine>` content. The annotated box is used only for target and loss construction and is never inserted into the model input.

Controlled-correction BDR state. For BDR, the shared corrupted anchor is represented exactly as

<refine> The target object is located at coordinates [$\mathbf{b}^{\text{in}}$]. </refine>
<answer>[$\mathbf{b}^{\text{in}}$]</answer>

There are no earlier numbered Steps. The two occurrences of $\mathbf{b}^{\text{in}}$ form the editable final-refinement and Answer roles, while the sentence, tags, and punctuation remain fixed. Autoregressive baselines receive the same image, expression, and corrupted anchor through the templates in Figure 7. No method receives the annotated box as input.

For an image of width W and height H, pixel coordinates are normalized as

$$x' = \text{round}(1000x/W), \qquad y' = \text{round}(1000y/H), \tag{42}$$

and clipped to $[0, 1000]$. RER and the native baseline use deterministic decoding for reported evaluation, while BDR applies digit-vocabulary argmax at the aligned coordinate positions. The evaluator parses one complete legal box from the final <answer> field. Invalid or unparseable predictions remain in the corresponding split denominator and receive zero IoU.

A.7 Implementation Notes

We implement RER and BDR on Qwen3.5-9B and train their adapters sequentially. RER and BDR initialize separate adapters, and standard evaluation uses one RER generation followed by two deterministic BDR passes. Complete optimization configurations, adapter settings, execution commands, and hardware details are provided in the anonymous supplementary code. All method-defining objectives, target rules, perturbation procedures, decoding protocols, and evaluation conventions are specified in the paper and appendices.

B Formal Properties

The following propositions characterize boundedness, normalization, gradient direction, and entropy gating in the implemented objectives.

Proposition B.1 (Bounded progressive signal) *The implemented progressive component satisfies* $0 \le C_{\text{prog}} \le 1$. *Consequently,* $0 \le R_{\text{prog}} \le R_{\text{fmt}}$.

Proof. Let $m_i^u = \max_{j \le i} u_j$ and $m_i^q = \max_{j \le i} q_j$. Every positive best-so-far increment telescopes, and therefore

$$\sum_{i=2}^{K} \frac{m_i^u - m_{i-1}^u}{1 - u_1 + \epsilon} \le 1, \qquad \sum_{i=2}^{K} \frac{m_i^q - m_{i-1}^q}{1 - q_1 + \epsilon} \le 1. \tag{43}$$

The non-positive increments retained in Eq. (23) can only decrease the sum. Its $0.6/0.4$ convex weighting and outer truncation thus give $C_{\text{prog}} \in [0, 1]$. The coverage, overall-gain, and consistency terms also lie in $[0, 1]$, and their coefficients in Eq. (25) sum to one. Multiplication by $R_{\text{fmt}} \in [0, 1]$ proves the second bound. □

Proposition B.2 (Normalized coordinate weighting) *For positive coordinate weights, the weighted box cross-entropy in Eq. (32) is invariant to a common positive scaling of all weights and remains between the minimum and maximum selected token losses.*

Proof. Writing c_t for per-token cross-entropy and $\alpha_t = w_t / \sum_j w_j$ gives $\mathcal{L}_{\text{box}} = \sum_t \alpha_t c_t$, where $\alpha_t > 0$ and $\sum_t \alpha_t = 1$. The loss is therefore a convex combination of the selected token losses. Replacing every w_t by λw_t for $\lambda > 0$ leaves each α_t unchanged, which proves both claims. □

Proposition B.3 (Preference-gradient direction) *For every active coordinate, gradient descent increases the output-versus-input margin for a beneficial edit and decreases it for a detrimental edit.*

Proof. Differentiating Eq. (37) yields

$$\frac{\partial \ell_{sk}}{\partial D_{sk}} = \sigma(D_{sk}) - y_{sk}. \tag{44}$$

If $y_{sk} = 1$, this derivative is negative and gradient descent increases D_{sk}. If $y_{sk} = 0$, it is positive and gradient descent decreases D_{sk}. Coordinates inside the dead zone $|\Delta_{sk}| \le 0.005$ are inactive and receive no preference gradient. □

Proposition B.4 (Correctness-gated entropy) *The final-coordinate entropy term cannot directly sharpen a digit whose digit-vocabulary argmax differs from the selected target digit.*

Proof. The gate includes a final digit position only when its current argmax equals the selected target. An incorrect position is therefore excluded from $\mathcal{H}_{\text{fin}}$ and has zero gradient from this term. Included positions minimize entropy only after satisfying the discrete correctness condition. The historical entropy term is ungated but contains no absolute coordinate target. □

Training and inference separation. The annotation $\mathbf{b}^*$ is used in training to select targets and compute edit labels. At inference, target selection, entropy gating, and preference labels are disabled; the model observes only (I, q) and the current response.

C Additional Experimental Results

C.1 Dataset Accounting

The archived localization evaluation contains the following eight standard splits. The high-precision, controlled-correction, and ablation studies use the five test splits shown in the final column.

Table 7: Dataset composition used by the archived evaluations.

Dataset	Split	Samples	Eight-split evaluation	Five-test evaluation
RefCOCO	val	10,834	✓	–
RefCOCO	testA	5,657	✓	✓
RefCOCO	testB	5,095	✓	✓
RefCOCO+	val	10,758	✓	–
RefCOCO+	testA	5,726	✓	✓
RefCOCO+	testB	4,889	✓	✓
RefCOCOg	val	4,896	✓	–
RefCOCOg	test	9,602	✓	✓
Total	eight / five splits	**57,457 / 30,969**		

For the main experiment, *Overall* is the arithmetic mean of the eight split-level $\mathrm{Acc@0.5}$ values. The RER aggregate in Table 8 is computed jointly over all 57,457 localization examples. Stricter-IoU, controlled-correction, meaningful-state ablation, and state-source aggregates are computed jointly over the 30,969 examples from the five test splits. The segmentation experiment uses 57,457 expression instances and retains this denominator separately from the localization tables.

C.2 RER Localization State Quality

Table 8 reports the terminal answer extracted from the RER state before BDR refinement. Unparseable predictions are assigned zero IoU and remain in the split denominators.

Terminal answer quality alone does not determine the final performance of a source-matched state–editor pipeline. Table 9 contrasts the native grounding output of Qwen3.5-9B, the same model under the structured RER interface without RER adaptation, the trained RER state, and refinement from the two state sources. The structured interface alone is substantially weaker than the native grounding interface, whereas RER adapts the policy to the explicit trajectory format and recovers most of the terminal localization quality. Each state source is paired with a separately trained BDR. The native-output pipeline starts at 82.16% terminal mIoU, whereas the RER pipeline starts at 80.70%. After source-matched refinement, RER + BDR gains 5.02 points to reach 85.72% mIoU and attains 68.64% $\mathrm{Acc@0.9}$. The native-output pipeline gains 2.41 points to reach 84.57% mIoU and attains 66.32% $\mathrm{Acc@0.9}$. This comparison highlights the effectiveness of the RER→BDR pairing: despite starting from a weaker terminal box, it obtains larger refinement gains and higher final localization quality. The zero-anchor ablation in Fig. 4 provides complementary evidence that a meaningful state and its anchor-aware objectives matter to the complete system.

Table 10 gives the split-level results of the meaningful-state ablation.

Table 8: RER terminal localization on the eight standard splits. The aggregate row is computed jointly across the splits.

Dataset	Split	Samples	mIoU	Acc@0.5	Acc@0.7	Acc@0.9
RefCOCO	val	10,834	83.35	90.10	83.81	63.61
RefCOCO	testA	5,657	84.09	90.75	85.56	64.27
RefCOCO	testB	5,095	81.37	88.38	80.73	59.02
RefCOCO+	val	10,758	77.14	82.57	76.15	57.82
RefCOCO+	testA	5,726	79.32	85.23	79.03	59.10
RefCOCO+	testB	4,889	74.18	80.18	72.20	53.12
RefCOCOg	val	4,896	82.49	89.22	82.15	60.72
RefCOCOg	test	9,602	82.48	89.28	82.70	61.45
Aggregate	**all splits**	**57,457**	**80.68**	**87.06**	**80.48**	**60.23**

Table 9: Terminal grounding and source-matched refinement under the five-test metric aggregation. Metrics are computed jointly across the splits; invalid or unparseable outputs receive zero IoU, and BDR rows use the second refinement pass. Each refined state source uses a separately trained BDR, and the native and structured rows use their corresponding prompts in Section A.6.

State source	Interface	RER adaptation	mIoU	Acc@0.7	Acc@0.9
Qwen3.5-9B	native output	No	82.16	83.31	60.64
Structured Qwen3.5-9B	structured	No	63.48	57.08	24.28
RER terminal answer (ours)	structured	Yes	80.70	80.56	59.81
Native-Output + BDR (ours)	native-output-derived	No	<u>84.57</u>	<u>86.56</u>	<u>66.32</u>
RER state + BDR (ours)	structured	Yes	**85.72**	**87.94**	**68.64**

Table 10: Split-level meaningful-state ablation. The upper block reports mIoU / Acc@0.5 and the lower block reports Acc@0.7 / Acc@0.9.

Method	RefCOCO A	RefCOCO B	RefCOCO+ A	RefCOCO+ B	RefCOCOg test
CoEvolve, mIoU / Acc@0.5	**88.85 / 95.28**	**85.41 / 92.38**	**87.12 / 93.45**	**81.23 / 87.79**	**85.47 / 92.75**
Zero-Anchor BDR, mIoU / Acc@0.5	<u>87.91 / 94.22</u>	<u>84.24 / 90.62</u>	<u>86.00 / 92.19</u>	<u>78.89 / 84.27</u>	<u>81.34 / 89.79</u>
CoEvolve, Acc@0.7 / Acc@0.9	**92.65 / 74.24**	**87.05 / 67.79**	**90.67 / 73.11**	**82.33 / 64.68**	**86.86 / 65.15**
Zero-Anchor BDR, Acc@0.7 / Acc@0.9	<u>90.26 / 69.91</u>	<u>84.02 / 65.32</u>	<u>87.91 / 67.69</u>	<u>77.79 / 60.20</u>	<u>81.97 / 53.62</u>

C.3 Qualitative Grounding Results

Figure 8 exposes one complete state transition. The RER response follows the required <refine> and <answer> format, and every reasoning Step terminates in an explicit region. BDR then edits coordinates at the same response positions: the coordinate matrix shows that refinement can update historical Steps, the last Step, and the Answer while retaining the surrounding reasoning text.

Selected localization states



RER structured assistant response

Query: "broccoli on top right"

<refine>

Step 1: The image shows a plate of food with a large piece of meat on the left, carrots in the center, and a pile of green vegetables on the right.
→ [0, 0, 1000, 1000]

Step 2: Focus on the right side of the plate where the green vegetables are located; these are the broccoli florets.
→ [600, 25, 1000, 650]

Step 3: Among the green vegetables, identify the ones positioned at the top right corner of the plate, which are the target broccoli pieces.
→ [600, 25, 1000, 450]

Step 4: Zoom in on the top right broccoli, excluding the lower florets and any other vegetables; this includes the two main florets at the top.
→ [605, 30, 1000, 420]

Step 5: Tightly enclose only the broccoli on the top right — the two distinct florets and their stems — trimming away all surrounding carrots and other vegetables.
→ [605, 30, 1000, 420]

</refine>

<answer>[605, 30, 1000, 420]</answer>

Same-position coordinate edits

Role	RER	BDR Pass 1	BDR Pass 2
Step 1	[0, 0, 1000, 1000]	**[0810, 0020, 1000, 0400]**	**[0810, 0020, 1000, 0420]**
Step 2	[600, 25, 1000, 650]	**[0810, 0025, 1000, 0400]**	**[0810, 0025, 1000, 0420]**
Step 3	[600, 25, 1000, 450]	**[0810, 0025, 1000, 0420]**	[0810, 0025, 1000, 0420]
Step 4	[605, 30, 1000, 420]	**[0815, 0030, 1000, 0420]**	**[0815, 0020, 1000, 0420]**
Step 5	[605, 30, 1000, 420]	**[0815, 0020, 1000, 0410]**	**[0815, 0020, 1000, 0420]**
Answer	[605, 30, 1000, 420]	**[0815, 0020, 1000, 0410]**	**[0815, 0020, 1000, 0420]**

Answer IoU: RER 0.451 → Pass 1 0.913 → Pass 2 0.936. Amber cells changed from the preceding state.

Figure 8: A structured RER trajectory and its subsequent BDR coordinate edits for the query "broccoli on top right." RER constructs a progressive localization trajectory, while BDR performs same-position coordinate refinement across multiple editable roles and substantially improves the final Answer. The image shows the first RER region, RER Answer, second-pass BDR Answer, and annotation. The lower matrix traces all coordinate states; amber cells mark edits from the preceding state.

Figure 9 visualizes additional RER-to-BDR localization updates on natural images. The cases are chosen to expose distinct correction behaviors rather than to estimate their frequency. Each panel preserves the full image, and its IoU is computed from the archived box shown in that panel.

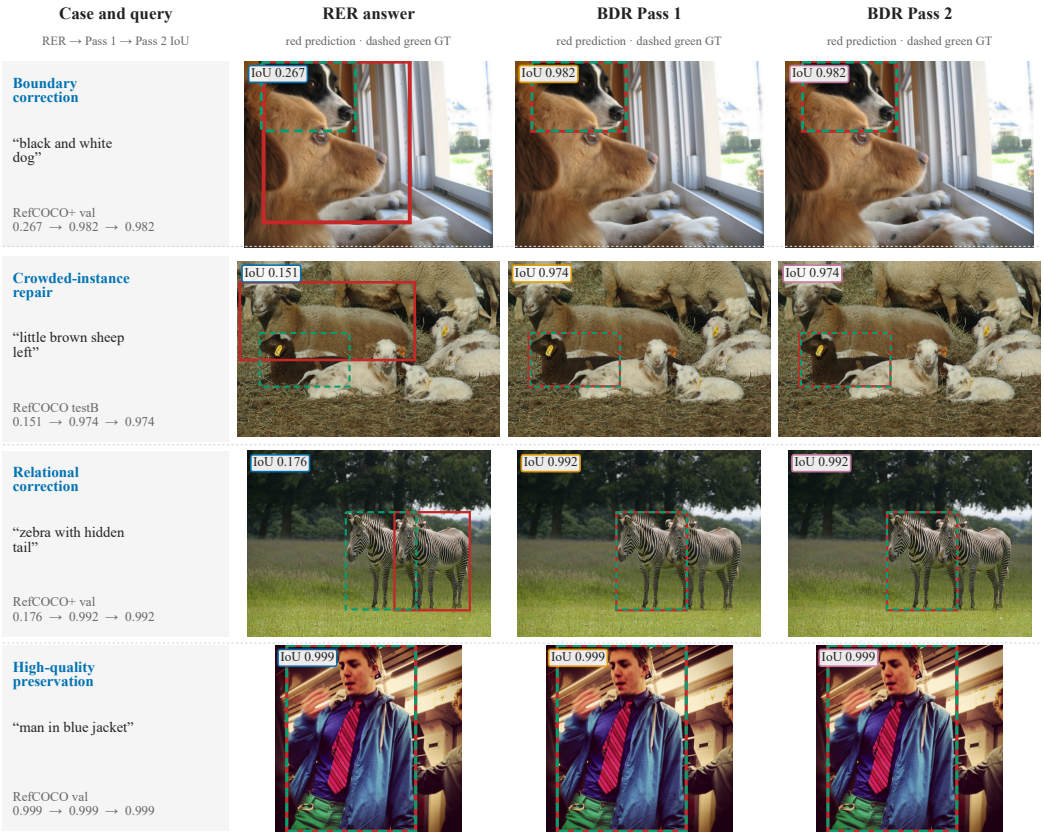

Figure 9: Qualitative results on the RefCOCO family. Red solid boxes show model predictions, green dashed boxes show annotations, and badges report per-example IoU. The examples illustrate boundary correction, disambiguation among nearby instances, relational correction, and preservation of an accurate RER anchor.

C.4 Controlled Correction Details

The controlled inputs are one deterministic nonzero perturbation per example, with target-IoU centers $0.20, 0.40, 0.60, 0.80, 0.94$ and four transformation types (translation, scale, aspect-ratio, and mixed). The same anchor is given to every model. CoEvolve uses one BDR pass in this experiment. Its Repair@0.7 is 87.99%, defined on the subset whose input IoU is below 0.7. Using the fixed seed 20260803, the construction procedure deterministically assigns each example a target-IoU band, perturbation type, and transformation parameters.

Figure 10 summarizes the benchmark construction. The five test splits contain 30,969 examples and have closely matched input mIoU values, yielding an aggregate input mIoU of 58.80 over all examples. The deterministic assignment is approximately uniform across the five target IoU bands and the four perturbation types, preventing the aggregate result from being dominated by one corruption severity or geometric transformation.

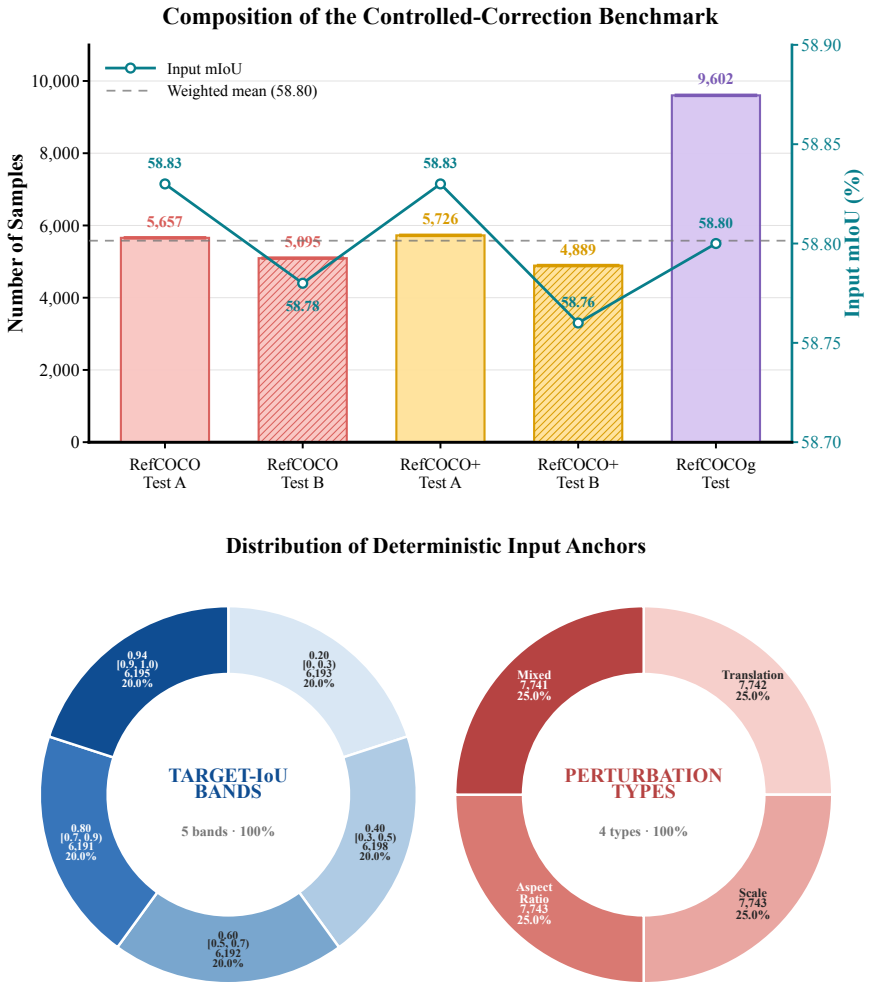

Figure 10: Construction of the controlled-correction benchmark. The upper panel reports the number of examples and input mIoU for each test split; the dashed line marks the aggregate input mIoU over all examples. The lower panel reports the assignment across target IoU bands and perturbation types. Each example contributes one deterministic nonzero anchor.

C.5 REMOTE-SENSING GROUNDING

DIOR-RSVG evaluation contains 7,500 expression instances. The main paper compares CoEvolve with prior methods, while Table 11 traces the three saved localization states. Unparseable outputs remain in the denominator and receive zero IoU. RER and BDR retain the natural-image reward, loss, and two-pass editing definitions while being retrained on the DIOR-RSVG training split. Their domain-specific optimization configurations are provided in the anonymous supplementary code. The reproduced native Qwen3.5-9B row in the main comparison is evaluated without DIOR-RSVG adaptation and is included as a base-to-system reference, not as a matched-training architectural comparison.

Table 11: CoEvolve localization-state progression on DIOR-RSVG test expressions. RER and BDR are retrained on the DIOR-RSVG training split.

State	mIoU	Pr@0.5	Pr@0.7	Pr@0.9
RER output	58.2248	64.2400	51.1200	23.0400
BDR Pass 1	75.4161	85.2533	74.6267	41.5200
BDR Pass 2	**76.2295**	**86.2000**	**76.1600**	**43.4533**

Figure 11 complements the aggregate progression with successful correction modes and a non-monotonic failure case.

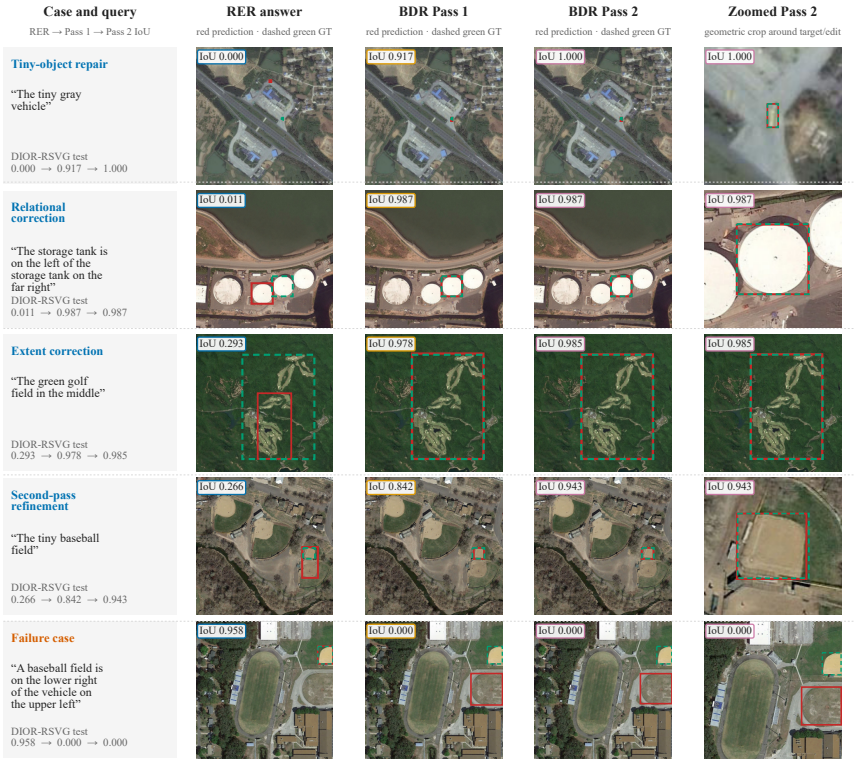

Figure 11: Qualitative results on DIOR-RSVG. Red solid boxes show model predictions and green dashed boxes show annotations. The first four rows show repair of a tiny object, a relationally specified instance, object extent, and a boundary further improved by the second BDR pass. The final row shows a failure in which refinement moves away from an initially accurate anchor. The rightmost column is a geometric crop around the annotation and final edit; no contrast enhancement is applied.

C.6 Downstream Segmentation

We feed each final localization box to a frozen SAM3 without mask fine-tuning. The segmentation archive contains 57,457 expression instances. Table 12 reports split-level cIoU and the arithmetic mean across the eight splits.

Table 12: Referring-expression segmentation cIoU on the eight standard splits. Public methods use their reported training and pretraining protocols; Macro-8 is the arithmetic mean of the eight split values. Best and second-best results are bold and underlined.

	RefCOCO			RefCOCO+			RefCOCOg		
Method	val	A	B	val	A	B	val	test	Macro-8
CRIS (Wang et al., 2022)	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	64.25
LAVT (Yang et al., 2022)	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	65.79
CGFormer (Tang et al., 2023)	74.75	77.30	70.64	64.54	71.00	57.14	64.68	65.09	68.14
ReLA (Liu et al., 2023a)	73.82	76.48	70.18	66.04	71.02	57.65	65.00	65.97	68.27
MagNet (Chng et al., 2024)	75.24	78.24	71.05	66.16	71.32	58.14	65.36	66.03	68.94
LISA (Lai et al., 2024)	74.90	79.10	72.30	65.10	70.80	58.10	67.90	70.60	69.85
PolyFormer-B (Liu et al., 2023b)	74.82	76.64	71.06	67.64	72.89	59.33	67.76	69.05	69.90
OneRef-L (Xiao et al., 2024)	80.48	82.78	78.27	74.25	78.41	69.85	74.91	77.36	77.04
EVF-SAM (Zhang et al., 2026a)	82.10	83.70	80.00	75.20	78.30	70.10	76.80	77.40	77.95
CoEvolve + SAM3 (ours)	80.28	82.81	77.87	76.22	80.60	72.42	77.06	78.23	78.19
LENS (Zhu et al., 2026a)	84.20	85.30	81.00	79.40	82.80	74.30	81.20	81.00	81.15
HyperSeg (Wei et al., 2025a)	84.80	85.70	83.40	79.00	83.50	75.20	79.40	78.90	81.24
InstructSeg (Wei et al., 2025b)	**85.80**	86.60	<u>84.00</u>	<u>80.10</u>	<u>83.80</u>	<u>75.60</u>	79.30	80.30	81.94
X-SAM (Wang et al., 2026b)	<u>85.10</u>	<u>87.10</u>	83.40	78.00	81.00	74.40	<u>83.80</u>	<u>83.90</u>	<u>82.09</u>
FlowSeg (Zhang et al., 2026b)	**85.80**	**87.40**	**84.80**	**80.20**	**84.50**	**76.90**	**86.50**	**86.10**	**84.03**

D Limitations

Imperfect preserve-or-edit decisions. BDR improves correction behavior on average but does not guarantee an improvement for every example. In controlled correction, refinement degrades 15.50% of the anchors, compared with 11.42% for Qwen3.5-9B. Quality-gated targets encourage preservation of accurate inputs, but the learned preserve-or-edit decision remains imperfect for a residual subset. Calibrated edit acceptance, uncertainty-aware refinement, or adaptive stopping could retain the current state when a further update is unlikely to help.

State–editor pairing and broader transfer. The source-matched comparison shows that RER→BDR achieves stronger final refinement than the separately trained native-output pipeline. This result motivates further study of how state representations interact with their corresponding editors, particularly across upstream model families and response formats.

Coordinate-centric editing. BDR treats reasoning text as fixed semantic context and edits the coordinate fields of the localization state. This design keeps the semantic context fixed during position-aligned refinement, but semantic errors that cannot be resolved through coordinate correction remain outside the current editor's scope. The current objective also does not explicitly enforce semantic-role consistency between an earlier reasoning step and its edited historical box. Extending bidirectional editing toward jointly editable semantic–spatial states is a promising direction.