MAKING LOCALIZATION THINKABLE: GROUNDING ACTIONS AS THOUGHT PRIMITIVES

Anonymous authors
Paper under double-blind review

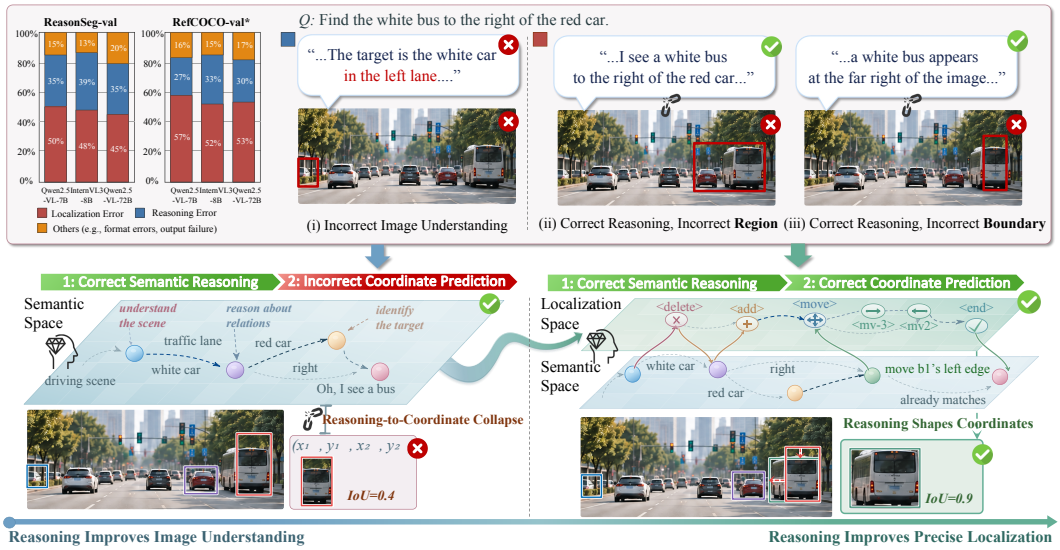

Figure 1: **Top**: Grounding error analysis reveals *Reasoning-to-Coordinate Collapse*: even with correct semantic reasoning, models often localize the incorrect region or produce inaccurate boundaries. **Bottom**: *ThinkAxis* brings localization into the reasoning process, enabling semantic understanding and coordinate formation to evolve together and allowing reasoning to shape precise localization.

ABSTRACT

Reasoning has become fundamental to modern Vision-Language Models (VLMs), extending visual understanding beyond direct prediction. In visual grounding, however, reasoning remains largely semantic, leaving precise localization outside the reasoning process. We term this gap *Reasoning-to-Coordinate Collapse*: VLMs can identify the correct target yet still predict inaccurate coordinates because localization is not explicitly modeled during reasoning. We present **ThinkAxis**, a paradigm that brings localization directly into the reasoning process. ThinkAxis represents localization updates as executable grounding primitives and carries the evolving localization throughout the reasoning trajectory, enabling VLMs to reason over and act on coordinates alongside semantic interpretation. To optimize this localization reasoning process, we further investigate the visual evidence behind localization decisions and find that accurate coordinate evolution increasingly relies on boundary-relevant evidence. Building on this observation, we formulate a trajectory optimization algorithm, E^2RPO, which couples geometric progress with counterfactual evidence attribution. Extensive experiments across visual grounding tasks show the effectiveness of ThinkAxis, with particularly notable improvements in precise localization under strict localization criteria. These results highlight the potential of localization-space reasoning for broader multimodal systems where precise spatial grounding is critical.

1 INTRODUCTION

Reasoning has become a defining capability of modern large models, enabling them to solve complex problems through iterative inference Wei et al. (2022); Guo et al. (2025). In VLMs, such reasoning

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.

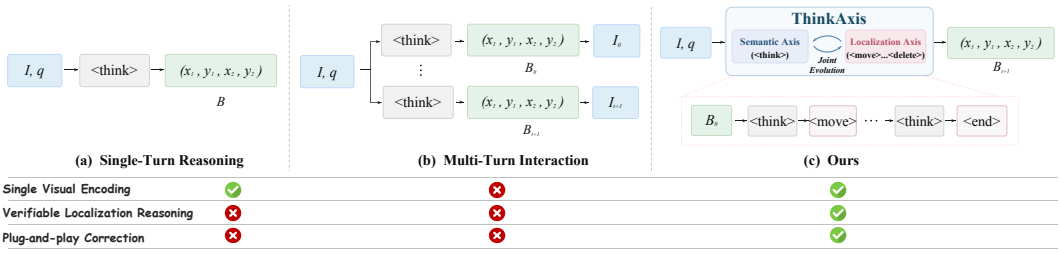

Figure 2: (a) Single-turn methods predict coordinates after semantic reasoning. (b) Multi-turn methods refine localization via visual feedback but keep coordinates implicit. (c) ThinkAxis models coordinate changes as explicit reasoning primitives within a unified trajectory.

extends visual understanding beyond direct prediction by progressively gathering and integrating visual evidence Qi et al. (2025); Lai et al. (2026); Hong et al. (2025). Visual grounding provides a natural setting for studying this capability, as it requires models to translate semantic understanding into precise spatial localization Wang et al. (2025b); Chen et al. (2023); You et al. (2023). However, existing grounding methods still reason primarily in the semantic space Bai et al. (2025c); Liu et al. (2025b;a), focusing on scene context, object attributes, and inter-object relations, while precise localization remains an implicit outcome rather than an explicit part of reasoning. *How reasoning can directly contribute to precise coordinate formation remains underexplored*, leaving a key gap between semantic reasoning and spatial localization.

To quantify this gap, we decompose grounding failures into two stages: semantic reasoning, which determines the referred target, and coordinate prediction, which maps the resolved target to precise image coordinates. Our error taxonomy in Fig. 1 reveals that many failures occur even when models correctly identify the target and reason consistently about it, yet still produce inaccurate coordinates. We term this phenomenon *Reasoning-to-Coordinate Collapse*: semantic reasoning succeeds but fails to translate into precise localization due to the lack of an explicit geometric anchor. Existing grounding methods mitigate such errors through multi-turn reasoning and visual interaction, progressively gathering target-relevant evidence Cao et al. (2025); Sun et al. (2026). However, as shown in Fig. 2, this reasoning still operates primarily in the semantic space, with coordinates produced only at the end of each reasoning round rather than being explicitly involved in the reasoning process. As a result, VLMs are effectively *thinking* in semantics while still *guessing* in pixels.

In this work, we rethink grounding reasoning by making localization an explicit part of the reasoning process: the model reasons not only about what the target is, but also about how its localization should evolve. We introduce **ThinkAxis**, a paradigm that augments the reasoning trace with an explicit localization axis, so that coordinate formation evolves alongside semantic interpretation rather than being deferred to the final prediction. This formulation raises three key challenges:

First, how does localization enter the reasoning process? Simply regenerating coordinates at each step does not explicitly represent how localization changes from one step to the next. In contrast, our analysis shows that relative updates yield more effective corrections by making two key geometric decisions explicit: direction and magnitude. This motivates us to treat localization updates as primitive units of spatial reasoning. We instantiate these primitives as executable grounding actions, including `<move>`, `<add>`, and `<delete>`. By turning coordinate updates into reasoning primitives, these actions allow the model to explicitly reason over and operate on coordinates, extending reasoning from semantic interpretation to the evolution of localization itself.

Second, how can localization and semantic reasoning be jointly evolved within a unified reasoning trajectory? A straightforward solution is multi-turn interaction, where each localization update triggers another round of semantic reasoning. However, such interaction relies on repeated visual interaction to drive localization evolution, introducing substantial computational overhead. More importantly, our analysis shows that longer interaction chains can diffuse visual attention and compound localization errors across rounds. With grounding actions represented in a unified token space, ThinkAxis instead composes successive localization updates into a single autoregressive trajectory under a shared visual context. This formulation preserves explicit localization evolution without repeated visual interaction, yielding a more compact reasoning process that improves both inference efficiency and localization accuracy.

Finally, how should the explicit localization trajectory be optimized? Because each grounding action has an explicit geometric effect, the resulting trajectory allows learning signals to be assigned throughout the localization process rather than only to the final prediction. Beyond geometric progress, we further observe that the visual evidence supporting localization changes systematically across stages: coarse localization relies more on object-interior cues, whereas precise correction increasingly depends on boundary evidence. Based on this observation, we extend GRPO Shao et al. (2024) with *edge-evidence attribution* to formulate E^2*RPO*, which couples route-level geometric utility with counterfactual boundary verification to optimize the reasoning-localization trajectory.

Extensive experiments show that ThinkAxis consistently improves visual grounding performance, with particularly pronounced gains under strict localization criteria, validating the effectiveness of incorporating localization into the reasoning process. We further show that the benefits of localization-aware reasoning extend to multi-turn visual understanding and localization-sensitive tasks in robotics and autonomous driving, where precise grounding is critical for reliable interaction and control.

Our contributions are summarized as follows:

- We identify *Reasoning-to-Coordinate Collapse*, revealing a fundamental gap between successful semantic reasoning and precise coordinate formation, and motivating localization itself to become part of the reasoning process.
- We introduce **ThinkAxis**, a paradigm that turns localization into an explicit evolving variable in reasoning by formulating grounding actions as thought primitives, enabling semantic reasoning and localization evolution to jointly progress along a unified reasoning trajectory.
- We uncover a shift in visual evidence from object-interior cues toward boundary evidence as localization evolves, and accordingly develop E^2*RPO* to jointly optimize spatial updates and action-relevant visual attribution via counterfactual verification.

2 Related Work

Reasoning-enhanced visual grounding. Inspired by Chain-of-Thought reasoning Wei et al. (2022), recent grounding methods increasingly introduce explicit reasoning to improve target understanding Xu et al. (2025); Zhang et al. (2023; 2024). R1-style approaches further elicit such reasoning through verifiable rewards Guo et al. (2025); Liu et al. (2026; 2025b); Shao et al. (2024); Liu et al. (2025a); Shen et al. (2025); Jiang et al. (2025); Bai et al. (2025c); Wang et al. (2025a), while methods such as DeepEyes Zheng et al. (2025) and VGR Wang et al. (2025b) introduce grounding or visual interaction to gather additional evidence. Existing methods typically follow a *reason-then-predict* paradigm, where localization remains a coordinate output after reasoning (Fig. 2(a)). ThinkAxis instead elevates coordinate changes into explicit, verifiable reasoning primitives, making localization an explicit axis of the reasoning trajectory.

Iterative localization. Another line of work improves grounding by iteratively updating localization. GETok Ren et al. (2026) models coarse-to-fine localization with relative spatial offsets, while Ground-R1 Cao et al. (2025) and IVT Tripathy & Krishnan (2026) feed predicted regions into subsequent reasoning through rendering or cropping; similar correction loops have also been explored in reasoning segmentation Du et al. (2026); Zhu et al. (2025b); He et al. (2025); Sun et al. (2026). These methods organize localization evolution as discrete interaction rounds, with updated predictions fed into subsequent reasoning or visual processing (Fig. 2(b)). ThinkAxis instead composes successive corrections within a single autoregressive trajectory over one visual encoding and can directly take an existing localization as input for plug-and-play correction (Fig. 2(c)).

Policy optimization for grounded reasoning. Recent policy optimization methods improve credit assignment beyond trajectory-level rewards by decomposing or propagating learning signals across intermediate reasoning steps Yu et al. (2026); Guo et al. (2026); Cui et al. (2025); Wang et al. (2026). For multimodal reasoning, PAPO Wang et al. (2026) and SRPO Jiang & Zou (2026) improve visual dependence through perception-aware objectives and token-level credit assignment, while Ground-R1 Cao et al. (2025) and ViGoRL Sarch et al. (2026) explicitly ground reasoning in predicted evidence regions or visual coordinates. However, these objectives largely treat visual evidence at a coarse level, without explicitly distinguishing the boundary cues that determine precise localization. E^2RPO addresses this gap by coupling geometric progress with counterfactual attribution to boundary-relevant evidence along the localization trajectory.

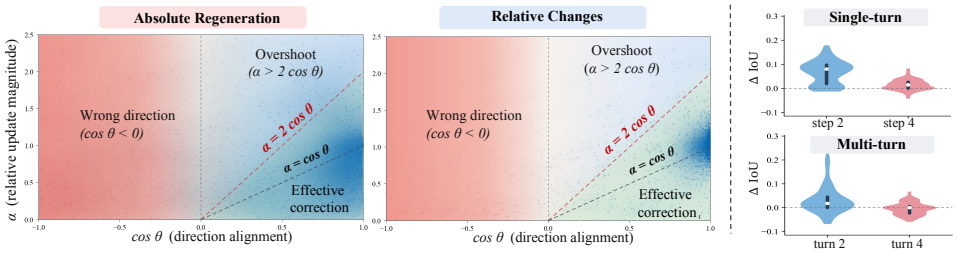

Figure 3: **Left:** Absolute regeneration vs. relative changes in direction alignment ($\cos\theta$) and update magnitude (α). **Right:** Violin plots show the distribution of ΔIoU for multi-turn vs. single-turn reasoning, with the width indicating sample density.

3 ThinkAxis: Localization Reasoning as Executable Actions

3.1 Preliminary Analysis: What Makes Localization Reasoning Effective?

In this work, we seek to understand how reasoning can improve localization toward precise coordinates. Human localization provides a useful intuition: precise boundaries are often established through successive corrections to a current spatial hypothesis. We therefore first train a *multi-turn grounding framework* that realizes this progressive localization process, revealing two key observations:

I. Relative changes provide a more explicit representation of localization evolution. Within this multi-turn framework, the next localization can be represented either as a regenerated absolute box B_{t+1} or as a relative update $\delta_t = B_{t+1} - B_t$. We compare the two by analyzing the geometry of the resulting update. Let $e_t = B^* - B_t$ denote the residual to the ground-truth box B^*. With e_t and δ_t normalized by the current box size, we define directional alignment $\cos\theta_t = \frac{\langle e_t, \delta_t \rangle}{\|e_t\|_2 \|\delta_t\|_2}$ and relative step size $\alpha_t = \frac{\|\delta_t\|_2}{\|e_t\|_2}$. The normalized reduction in localization error is

$$G_t = \cos^2\theta_t - (\alpha_t - \cos\theta_t)^2. \tag{1}$$

This decomposition makes the two geometric decisions explicit: aligning the update with the residual and matching its step size accordingly. As shown in Fig. 3, in the $(\cos\theta_t, \alpha_t)$ space, relative prediction yields more high-gain updates than absolute regeneration, suggesting that directly modeling the localization change reduces the coupling between these two decisions.

II. Repeated visual encoding brings limited additional gains to localization evolution. Beyond the update representation, multi-turn grounding also couples the evolution of visual observation and localization. We decompose each round into

$$I_{t+1} = \mathcal{R}(I_t, B_t), \qquad B_{t+1} = \mathcal{U}(I_{t+1}, q, B_t), \tag{2}$$

where $\mathcal{R}$ refreshes the visual observation and $\mathcal{U}$ updates the current localization. By isolating these two processes, we find that substantial localization gains are retained when I_t remains fixed while B_t continues to evolve. As shown in Fig. 3, our analysis of the two reasoning paradigms further shows that multi-turn reasoning exhibits a higher proportion of unintended erroneous corrections at later turns, suggesting reduced reliability over longer localization reasoning chains. This suggests that localization evolution can be separated from repeated observation refresh and be composed under a shared visual context. Together, these observations shape ThinkAxis as explicit relative localization transitions over a single visual encoding.

3.2 Problem Formulation

ThinkAxis augments semantic reasoning with an explicit localization axis, along which the current localization evolves throughout the reasoning trajectory. At step t, the preceding reasoning $\tau_{<t}$ and current localization B_t jointly determine a localization action

$$a_t \sim \pi_\theta(\cdot \mid I, q, B_t, \tau_{<t}), \qquad B_{t+1} = \mathcal{E}(B_t, a_t), \quad a_t \in \mathcal{A}_{\text{loc}}, \tag{3}$$

where a_t explicitly describes how the localization should evolve and $\mathcal{E}$ executes this transition in coordinate space. In this way, semantic interpretation and coordinate formation evolve within the same reasoning process as shown in Fig 4.

Localization reasoning primitives. We instantiate the localization axis with $\mathcal{A}_{\text{loc}} = \mathcal{A}_{\text{move}} \cup \mathcal{A}_{\text{add}} \cup \mathcal{A}_{\text{delete}} \cup \mathcal{A}_{\text{end}}$. <move> adjusts the current box boundaries, <add> starts a new localization route from the backbone's native box representation, <delete> removes the current localization, and <end> terminates the route. For <move>, we define a movement vocabulary $\mathcal{V}_{\text{mv}} = \{\texttt{<mv-}K\texttt{>}, \ldots, \texttt{<mv0>}, \ldots, \texttt{<mv}K\texttt{>}\}$. Its displacement levels are non-uniformly quantized as $\Psi = \{0, \pm\epsilon, \pm 2\epsilon, \ldots, \pm 2^{K-1}\epsilon\}$ ($\epsilon = 0.003$). A <move> selects one level for each boundary, scaled by the corresponding box width or height. This non-uniform spacing provides fine resolution near the target while preserving a wide correction range with few tokens, reducing the action space and facilitating more efficient optimization.



Figure 4: **Examples of ThinkAxis reasoning.** Semantic and localization thoughts jointly improve target localization.

Data curation. Existing grounding CoT data provide little guidance on *how* an imperfect localization should evolve. To bridge this gap, we develop an automated pipeline that converts standard grounding annotations into localization-space reasoning trajectories. For each sample, we first obtain initial localizations from a frozen grounding model, providing realistic starting points that reflect inference-time errors. Each prediction is matched to the ground-truth instance with the highest IoU. Predictions with very low overlap are replaced through a <delete>-<add> route. For others, each box boundary is moved toward its target boundary using the largest available displacement <mv*>. Finally, a multimodal teacher annotates concise <think> spans conditioned on the current localization and the prescribed next action.

3.3 E^{2}RPO: Edge-Evidence Relative Policy Optimization

Supervised fine-tuning teaches the model to imitate predefined correction trajectories, yet the model cannot verify whether its own generated actions actually improve localization. We therefore propose E^{2}RPO to optimize ThinkAxis through reinforcement learning, enabling the policy to actively explore and act in the 2D localization space under fine-grained geometric and edge-evidence feedback.

3.3.1 Reward Design

Localization reward. For each input $x_i = (I_i, q_i)$ with ground-truth box set $\mathcal{B}_i^*$, the model generates K candidate reasoning paths indexed by j, each consisting of an initial localization followed by a sequence of executable spatial actions. Executing these actions yields the initial and terminal predictions $\widehat{\mathcal{B}}_{i,j}^0$ and $\widehat{\mathcal{B}}_{i,j}^T$. To measure localization quality, we define

$$Q(\mathcal{B}, \mathcal{B}^*) = \frac{1}{|\mathcal{H}|} \sum_{\eta \in \mathcal{H}} \frac{\sum_{(m,n)\in M} \mathbb{I}\left[\text{IoU}(B_m, B_n^*) \geq \eta\right]}{\max(|\mathcal{B}|, |\mathcal{B}^*|)}, \qquad \mathcal{H} = \{0.5, 0.75, 0.9\}, \tag{4}$$

where M denotes the maximum IoU matching between the predicted and ground-truth boxes. The initial and terminal predictions are rewarded as $R_{i,j}^0 = Q(\widehat{\mathcal{B}}_{i,j}^0, \mathcal{B}_i^*)$ and $R_{i,j}^T = Q(\widehat{\mathcal{B}}_{i,j}^T, \mathcal{B}_i^*)$.

Process reward. We assign each target route a geometric utility $U_{i,j,r}$ according to its localization effect. <add> and <delete> are scored by localization correctness, while <move> considers both boundary-wise direction and realized progress:

$$U_{i,j,r}^{\text{move}} = \lambda_{\text{dir}} D_{i,j,r} + \lambda_{\text{prog}} \Phi_{i,j,r}, \qquad \Phi_{i,j,r} \in \{0, 0.25, 0.5, 1\}. \tag{5}$$

where $D_{i,j,r}$ measures boundary-wise directional correctness and $\Phi_{i,j,r}$ quantizes the realized localization progress. The process reward averages $U_{i,j,r}$ over all target routes, assigning zero utility to unresolved routes.

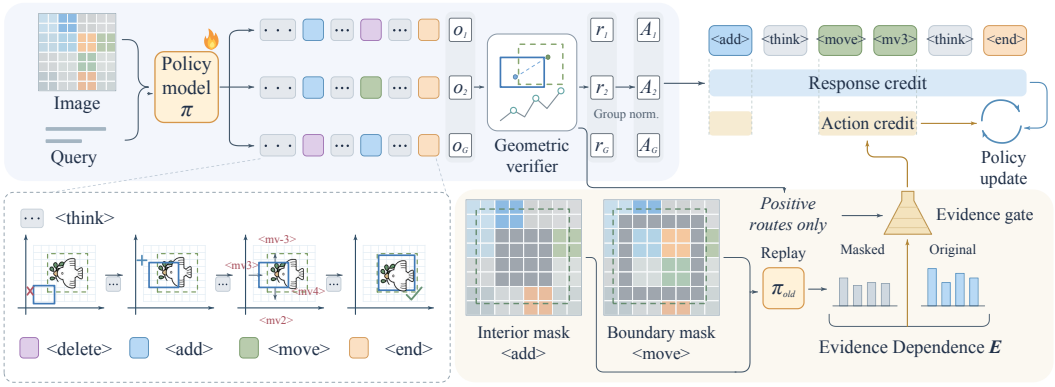

Figure 5: **ThinkAxis optimization with E^{2}RPO.** Two-level credit assignment guides the trajectory optimization: response-level credit evaluates the overall localization trajectory, while action-level credit couples geometric progress with evidence-aware signals via counterfactual attribution.

Overall reward. Let $F_{i,j} \in \{0,1\}$ denote format validity. Combining endpoint localization quality with process-level utility, the overall reward for rollout j is defined as

$$R_{i,j} = F_{i,j}\left(1 + \lambda_0 R^0_{i,j} + \lambda_{\text{final}} R^T_{i,j} + \lambda_{\text{proc}} R^{\text{proc}}_{i,j}\right), \tag{6}$$

3.3.2 Route-Level Relative Advantage

Although the overall reward evaluates both endpoint quality and process utility, standard GRPO still assigns a single group-relative advantage to all tokens in a rollout. This is too coarse for ThinkAxis, where different localization routes may contribute very differently to the final prediction. Route-level credit is therefore computed directly from the geometric utility $U_{i,j,r}$. For prompt i, the utilities of all active routes across the K rollouts are pooled as

$$\mathcal{P}_i = \left\{U_{i,j,r} \;\middle|\; j \in \{1,\dots,K\},\ r \in \mathcal{R}^{\text{act}}_{i,j}\right\}, \qquad A^{\text{route}}_{i,j,r} = \frac{U_{i,j,r} - \mu(\mathcal{P}_i)}{\sigma(\mathcal{P}_i) + \epsilon}. \tag{7}$$

The route-level advantage is applied only to the corresponding localization-action tokens, while semantic reasoning tokens retain the response-level GRPO advantage. This separates credit for individual spatial decisions from the global quality of the complete reasoning trajectory.

3.3.3 Counterfactual Edge-Evidence Attribution

Route-level advantage captures the geometric utility of each localization action. However, localization actions may rely on different visual evidence. Our analysis reveals that accurate corrections exhibit stronger dependence on target-edge evidence, motivating an evidence-aware optimization signal.

To estimate such dependence, we fix the generated trajectory and measure the policy response difference between the original image I_i and a counterfactual image $\widetilde{I}_{i,j}$, where action-relevant evidence is masked. Specifically, target-interior regions are masked for credited `<add>`, while a narrow target-edge band is masked for other eligible correction actions. For efficiency, the selected regions within each rollout are merged into a single counterfactual image. Let $\mathcal{T}_{i,j,r}$ denote the action-token positions of route r. Its evidence dependence is measured by the drop in action log-likelihood:

$$E_{i,j,r} = \frac{1}{|\mathcal{T}_{i,j,r}|} \sum_{t \in \mathcal{T}_{i,j,r}} \left[\log \pi_{\text{old}}(y_t \mid I_i, q_i, y_{<t}) - \log \pi_{\text{old}}(y_t \mid \widetilde{I}_{i,j}, q_i, y_{<t})\right]. \tag{8}$$

A larger $E_{i,j,r}$ means that removing the relevant visual evidence causes a larger confidence drop for the same localization action, indicating stronger evidence dependence. Evidence dependence is meaningful only when the corresponding correction is geometrically useful. It therefore provides additional credit only to routes with positive utility and positive relative advantage:

$$A^{\text{E}^2}_{i,j,r} = \left[A^{\text{route}}_{i,j,r}\right]_+ \mathbb{1}[U_{i,j,r} > 0]\,(g(E_{i,j,r}) - 1), \tag{9}$$

where $g(E) \geq 1$ is a bounded nondecreasing mapping and 1 denotes no additional evidence credit. The resulting bonus is applied only to the corresponding action tokens.

Table 1: **Referring expression comprehension at standard and stricter IoU thresholds.** (a) Acc@0.5 (%) on standard splits, with Avg. over eight splits. (b) Split-averaged accuracy at stricter IoU thresholds. ThinkAxis uses the final localization.

Method (Acc@0.5)	RefCOCO			RefCOCO+			RefCOCOg		Avg.
	Val	TestA	TestB	Val	TestA	TestB	Val	Test	
Grounding DINO-L Liu et al. (2024)	90.6	93.2	88.2	82.8	89.0	75.9	86.1	87.0	86.6
Ground-R1 Cao et al. (2025)	92.9	93.9	88.0	86.5	90.8	78.8	90.1	90.2	88.9
GETok Ren et al. (2026)	90.9	93.6	88.2	87.1	90.8	78.9	89.9	91.2	88.8
DiG-8B Tao et al. (2026)	90.0	91.9	86.4	84.2	87.7	77.6	87.7	88.0	86.7
VisionReasoner† Liu et al. (2026)	89.1	91.0	87.0	85.0	87.6	78.8	87.6	88.5	86.8
Qwen2.5-VL-7B Bai et al. (2025b)	90.0	92.5	85.4	84.2	89.1	76.9	87.2	87.2	86.6
ThinkAxis-7B	92.4	94.5	87.8	88.0	92.3	81.6	88.9	88.2	89.2
Qwen3-VL-8B Bai et al. (2025a)	92.0	93.6	87.9	86.6	91.2	80.7	88.8	88.8	88.7
ThinkAxis-8B	**94.8**	**95.1**	**90.6**	**89.5**	**92.4**	**83.6**	**90.0**	**90.3**	**90.8**
Split-Avg. Acc@IoU	@0.5	@0.7	@0.9	@0.5	@0.7	@0.9	@0.5	@0.7	@0.9
Qwen2.5-VL-7B† Bai et al. (2025b)	89.3	80.0	53.5	83.4	73.7	49.6	87.2	76.2	52.0
ThinkAxis-7B	91.6	82.8	60.1	87.3	77.1	56.4	88.6	80.8	58.2
Δ (vs. Qwen2.5-VL-7B)	**+2.3**	**+2.8**	**+6.6**	**+3.9**	**+3.4**	**+6.8**	**+1.4**	**+4.6**	**+6.2**

Table 2: **Performance on segmentation.** Precise localization transfers to dense mask.

Method	ReasonSeg		RefC	RefC+	RefCg
	Val.	Test	TestA	TestA	Test
Seg-R1 You & Wu (2025)	58.6	56.7	78.7	70.9	71.4
SAM-R1 Huang et al. (2026)	64.0	60.2	79.2	74.7	73.1
VisionReasoner Liu et al. (2026)	**66.3**	63.6	79.3	72.2	72.2
ThinkAxis	65.8	**64.6**	**80.3**	**77.2**	**74.6**

Table 3: **Performance on RealVLG.** RealVLG targets robotic manipulation requiring precise localization.

Method	Seen		Similar		Novel	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
Gemini2.5-Flash Comanici et al. (2025)	60.1	61.5	59.2	60.5	62.6	63.7
Qwen2.5-VL-7B Bai et al. (2025b)	56.8	58.3	57.3	57.9	55.6	56.8
RealVLG-R1 Li et al. (2026)	88.0	88.5	83.4	84.2	84.6	85.4
ThinkAxis	**89.3**	**89.6**	**88.0**	**88.5**	**85.1**	**85.8**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmark settings. We evaluate ThinkAxis from three perspectives. *(i) Precise localization* assesses whether localization-space reasoning improves coordinate formation through referring expression comprehension (REC) and referring expression segmentation (RES). *(ii) Reasoning enhancement* integrates ThinkAxis into DeepEyes Zheng et al. (2025) to investigate whether localization reasoning can further improve visual reasoning beyond grounding itself. *(iii) Real-world spatial understanding* evaluates ThinkAxis on RealVLG Li et al. (2026) and autonomous-driving scenarios, where precise spatial grounding is essential for downstream perception and decision-making.

Training details. ThinkAxis is built upon Qwen2.5-VL-7B and Qwen3-VL-8B and trained on NVIDIA A800 80GB GPUs. Training consists of two stages. For supervised fine-tuning, reasoning trajectories are automatically constructed from box annotations through our data pipeline. The tokenizer vocabulary is extended with localization-action tokens, with token embeddings and the LM head kept trainable. LoRA Hu et al. (2021) is applied with $r = 64$ and $\alpha = 128$. During reinforcement learning, E^{2}RPO is implemented using `verl` Sheng et al. (2024), using a batch size of 16 and 8 rollouts per prompt. We use a learning rate of 10^{-6}, weight decay of 0.01, and KL coefficient of 0.01. The edge-evidence coefficient is linearly warmed up during the first 100 steps.

4.2 MAIN RESULTS

Precise grounding and segmentation transfer. As shown in Tab. 1, ThinkAxis consistently improves REC Benchmarks, achieving average gains of **+2.6%** over Qwen2.5-VL-7B and **+2.1%** over Qwen3-VL-8B under Acc@0.5. Under the stricter Acc@0.9 criterion, the average gain further increases to **+6.5%**, indicating a stronger improvement in precise localization. The larger gains under high IoU suggest that explicitly modeling localization evolution helps VLMs better form accurate spatial boundaries, beyond merely identifying the referred target. Furthermore, this improved localization capability transfers naturally to dense segmentation: as shown in Tab. 2, directly using predicted boxes as prompts for SAM Kirillov et al. (2023) achieves competitive performance without task-specific training or modifications to the output format.

Enhancing tool-augmented visual reasoning. Beyond improving grounding itself, we investigate whether explicit localization reasoning can enhance broader visual reasoning. As shown in Tab. 4,

Table 4: **ThinkAxis for tool-augmented visual reasoning.** ThinkAxis can be integrated into tool-interleaved visual reasoning through either task-specific training or lightweight tool-based correction.

Method	V^* Bench			HR-Bench 4K			HR-Bench 8K		
	Attr.	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall
DeepEyes Zheng et al. (2025)	91.3	85.5	89.0	90.8	56.0	73.4	85.3	54.3	69.8
(a) + **ThinkAxis** (Training)	91.3	**88.2**	90.1	90.5	**56.8**	73.6	**86.5**	**57.5**	**72.0**
(b) + **ThinkAxis** (Corrector)	**92.2**	**88.2**	**90.6**	**91.3**	56.8	**74.1**	85.8	56.3	71.0

Table 5: **Real-world driving case study.** We evaluate ThinkAxis in challenging real-world traffic scenes, where visual targets are often small, distant, and densely distributed.

Method	IoU@0.25			IoU@0.50			IoU@0.75			mIoU	Count
	P	R	F1	P	R	F1	P	R	F1		
Qwen3-VL-8B	66.97	45.45	54.15	57.51	39.03	46.50	33.23	22.55	26.87	37.36	33.73
w. GRPO	76.05	**79.48**	77.73	72.25	**75.51**	73.84	57.69	**60.29**	58.96	57.44	37.83
ThinkAxis	**87.43**	72.35	**79.18**	**84.28**	69.74	**76.32**	**71.10**	58.84	**64.39**	**61.78**	**46.02**

we integrate ThinkAxis into DeepEyes in two settings. In the setting (a), ThinkAxis is first trained with a dedicated localization reasoning stage before the DeepEyes pipeline, enabling localization reasoning to be incorporated into subsequent visual reasoning. This integration improves HR-Bench 8K by **+2.2**. Moreover, when applied as a plug-and-play corrector to a frozen DeepEyes model (b), ThinkAxis further improves V^* Bench by **+1.6**, demonstrating its potential as a generalizable localization component for existing visual reasoning agents without additional retraining.

Generalization to real-world manipulation grounding. Precise localization is particularly critical in robotic manipulation scenarios, where grounding errors can directly affect downstream interactions with objects. We therefore evaluate ThinkAxis on RealVLG, a real-world grounding benchmark that measures localization performance under diverse scenarios. Using only **1/3** of the original training data, ThinkAxis consistently outperforms RealVLG-R1 across all settings, achieving the largest improvements with gains of **+4.6 gIoU** and **+4.3 cIoU**. These results demonstrate the potential of ThinkAxis for achieving precise localization under demanding real-world scenarios.

Real-world driving case study. We deploy ThinkAxis in a challenging autonomous-driving annotation setting, focusing on traffic-sign grounding where targets are small, distant, and densely distributed, making precise localization difficult. Evaluated on a real-world annotation set with 2,805 images and 11,131 annotated objects, ThinkAxis outperforms the GRPO baseline following the VisionReasoner Liu et al. (2026) training recipe, achieving gains of up to **+5.43 F1** and **+4.34 mIoU**. These results demonstrate the potential of localization evolution for generating high-quality annotations by enabling box correction and redundant prediction removal.

4.3 Ablation Studies and Analysis

Representing localization primitives. An effective correction requires both a correct direction ($\cos\theta > 0$) and an appropriate magnitude ($\alpha < 2\cos\theta$), with the maximum geometric gain attained at $\alpha = \cos\theta$. As shown in Fig. 3, offset prediction produces more effective localization updates, motivating explicit modeling of localization changes. We further observe that raw pixel offsets frequently degenerate into negligible adjustments, such as $+1$ pixels, which contribute little localization gain but add extra reasoning steps. To this end, we discretize relative displacements into structured tokens, forming a compact action space. The analysis is provided in Appendix B.4.

Single-trajectory vs. multi-turn localization. As shown in Fig. 3, multi-turn localization provides early improvements, but later updates gradually become negligible or even counterproductive. We further observe that longer interaction chains exhibit progressive attention diffusion, consistent with prior observations of perceptual degradation in extended multimodal reasoning Peng et al. (2026). In contrast, ThinkAxis maintains a shared visual context throughout localization evolution. On RefCOCOg, this single-trajectory design improves $\mathrm{Acc}@0.9$ by **+3.1%** while reducing inference latency by **46%**, demonstrating that this design enables both more precise and efficient grounding.

Visual evidence for precise localization. We analyze the visual evidence behind different localization actions following Eq. 8. By masking the target boundary and interior region, we compute E^{bd} and E^{in}, and define the causal gap as $g = E^{\mathrm{bd}} - E^{\mathrm{in}}$. Positive g indicates stronger dependence on

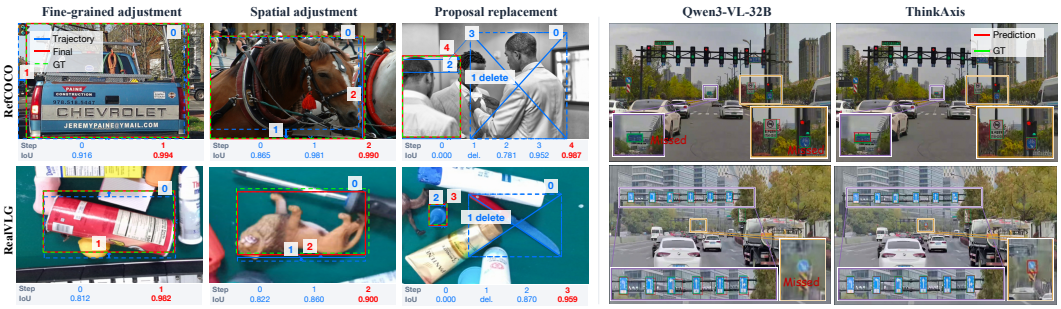

Figure 6: **Qualitative results.** ThinkAxis corrects both imprecise boundaries and incorrect target selections (**left**), while reducing missed targets and improving box alignment in driving scenes (**right**).

Table 6: **Effect of evidence attribution.** Higher causal gap indicates stronger boundary dependence.

Optimization	Causal Gap ↑	Outcome Gap ↑
w/o evidence credit	10.92	10.54
Per-route	**17.52**	10.87
Sample-union	15.82	**14.60**

Table 7: **Localization behavior across stages.** Corrections become smaller and reliable near the target.

Current IoU	Dir. Acc.	$\|\delta\|_{\text{rel}}$ (%)	$\mathbf{P}(\mathbf{G} > \mathbf{0})$ ↑
$[0.3, 0.5)$	85.0	11.44	86.7
$[0.5, 0.7)$	86.9	9.18	96.6
$[0.7, 0.9)$	86.9	5.13	97.5
$[0.9, 0.95)$	85.4	1.54	**100.0**

boundary evidence. Proposal actions (<box>/<add>) exhibit an average gap of -1.11, while <move> reaches $+10.92$ (both $p < 0.001$), revealing a clear transition from interior to boundary evidence during localization evolution. Furthermore, successful corrections show stronger boundary dependence, with outcome gaps of up to 14.60 under Sample-union attribution.

Optimization dynamics. Fig. 7 compares optimization strategies under Acc@0.9. The three variants show similar early learning trends, but full E^{2}RPO maintains higher accuracy in later training steps. Removing edge-evidence attribution weakens this advantage and does not consistently outperform standard GRPO, indicating that fine-grained geometric credit alone is insufficient. These results suggest that once basic grounding capability is established, boundary-focused reasoning becomes critical for precise localization.

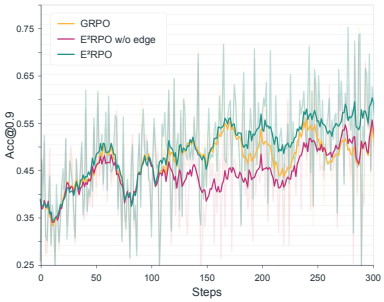

Figure 7: **Optimization dynamics** for precise localization.

Behavior of localization reasoning. To understand the bottleneck of precise correction, we examine how movement direction and magnitude evolve across localization stages in Tab. 7. Directional accuracy remains high across IoU ranges, indicating that direction selection becomes reliable relatively early. As localization improves, the remaining gains come primarily from better magnitude ($|\delta|_{\text{rel}}$) calibration, particularly by avoiding overshooting near the target.

Qualitative results. Fig. 6 further illustrates how semantic and localization reasoning jointly evolve in ThinkAxis. When the target is correctly understood, localization actions progressively adjust the box from a coarse prediction to a precise boundary. When the target is ambiguous, semantic reasoning guides localization actions to remove the incorrect box and search for a new location. Driving examples further highlight ThinkAxis in challenging scenes, where one-shot prediction often misses targets or produces inaccurate boundaries. ThinkAxis instead reasons over and repairs incomplete localization hypotheses, progressively improving target coverage and boundary precision.

5 Conclusion

This work revisits visual grounding from the perspective of reasoning, showing that precise localization requires more than semantic understanding alone. By introducing ThinkAxis, we demonstrate that localization can be progressively formed, corrected, and optimized within a reasoning trajectory through explicit spatial actions. The consistent improvements across grounding, visual reasoning, and real-world scenarios suggest that integrating task-critical variables into the reasoning process is a promising direction for building more capable multimodal systems.

AI USE STATEMENT

We used AI tools to assist with manuscript drafting, language polishing, and code revision and debugging. The authors take full responsibility for the final content of this work, including the methodology, data, code, figures, and reported results.

ETHICS STATEMENT

This work studies localization reasoning for vision-language models using publicly available visual grounding benchmarks. All datasets used in our experiments are released for research purposes and do not contain personally identifiable information collected by the authors. Our method does not involve human subjects, private data collection, or any additional annotation that introduces privacy concerns. The proposed capability may benefit applications requiring precise spatial understanding, such as robotics and assistive systems. However, improved visual localization capabilities may also raise concerns when applied to sensitive scenarios such as surveillance. We acknowledge these risks and emphasize that our work is intended solely for academic research and beneficial applications. No human subjects, personally identifiable information, or harmful content were involved in this study. We believe the ethical risks of this work are minimal and manageable.

REPRODUCIBILITY STATEMENT

We are committed to reproducible research and will release the implementation of ThinkAxis, including the trajectory construction pipeline, supervised fine-tuning framework, and reinforcement learning training procedure upon publication. The paper and appendix provide detailed descriptions of our model architecture, training configurations, data construction process, reward formulations, and evaluation protocols. We report the backbone models, optimization settings, and hyperparameters for both supervised fine-tuning and E^2RPO training to facilitate independent reproduction. Experiments are conducted on NVIDIA A800 GPUs. We believe the provided implementation details and experimental configurations enable reliable verification of our results.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *arXiv preprint arXiv:2505.14231*, 2025c.

Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. Ground-r1: Incentivizing grounded visual reasoning via reinforcement learning. *arXiv preprint arXiv:2505.20272*, 2025.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv:2306.15195*, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Tianyuan Du, Haopeng Li, Zhen Fan, Jiarui Zhang, Panwang Pan, and Yang Zhang. Sam-veteran: An mllm-based human-like sam agent for reasoning segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *Advances in Neural Information Processing Systems*, 38:114399–114431, 2026.

Xingqi He, Yujie Zhang, Shuyong Gao, Wenjie Li, Lingyi Hong, Mingxi Chen, Kaixun Jiang, Jiyuan Fu, and Wenqiang Zhang. Rsagent: Learning to reason and act for text-guided segmentation via multi-turn tool invocations. *arXiv preprint arXiv:2512.24023*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021.

Jiaqi Huang, Zunnan Xu, Jun Zhou, Ting Liu, Yicheng Xiao, Mingwen Ou, Bowen Ji, Xiu Li, and Kehong Yuan. Sam-r1: Leveraging sam for reward feedback in multimodal segmentation via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:138362–138383, 2026.

Bingqing Jiang and Difan Zou. Structured role-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2605.07274*, 2026.

Qing Jiang, Xingyu Chen, Zhaoyang Zeng, Junzhi Yu, and Lei Zhang. Rex-thinker: Grounded object referring via chain-of-thought reasoning. *arXiv preprint arXiv:2506.04034*, 2025.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pp. 4015–4026, 2023.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In *International Conference on Learning Representations*, volume 2026, pp. 76722–76746, 2026.

Linfei Li, Lin Zhang, and Ying Shen. Realvlg-r1: A large-scale real-world visual-language grounding benchmark for robotic perception and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pp. 38–55, 2024.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025a.

Yuqi Liu, Tianyuan Qu, Zhisheng Zhong, Bohao Peng, Shu Liu, Bei Yu, and Jiaya Jia. Visionreasoner: Unified reasoning-integrated visual perception via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 94069–94086, 2026.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2034–2044. IEEE, 2025b.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pp. 11–20, 2016.

Ruiying Peng, Xueyu Wu, Jing Lei, Lu Hou, Yuanzheng Ma, and Xiao-Hui Li. Deeper thought, weaker aim: Understanding and mitigating perceptual impairment during reasoning in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12064–12073, 2026.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Xu Bin, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: A visual language model with chain-of-manipulations reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 11090–11110, 2025.

Xiangxuan Ren, Zhongdao Wang, Liping Hou, Pin Tang, Guoqing Wang, and Chao Ma. Grounding everything in tokens for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41171–41181, 2026.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. *Advances in Neural Information Processing Systems*, 38:150977–151013, 2026.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model. *arXiv preprint arXiv:2504.07615*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Wenfang Sun, Hao Chen, Yingjun Du, Yefeng Zheng, and Cees G Snoek. Regionreasoner: Region-grounded multi-round visual reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 80583–80603, 2026.

Zhou Tao, Shida Wang, Yongxiang Hua, Haoyu Cao, and Linli Xu. Dig: differential grounding for enhancing fine-grained perception in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1695–1705, 2026.

Animesh Tripathy and Aswanth Krishnan. Iterative visual thinking: Teaching vision-language models spatial self-correction through visual feedback. *arXiv preprint arXiv:2606.13156*, 2026.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025a.

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, et al. Vgr: Visual grounded reasoning. *arXiv preprint arXiv:2506.11991*, 2025b.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multimodal reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 80502–80530, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *ICCV*, pp. 2087–2098, 2025.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv:2310.07704*, 2023.

Zuyao You and Zuxuan Wu. Seg-r1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *ECCV*, pp. 19–35, 2024.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv: 2302.00923*, 2023.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025a.

Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. Segagent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *CVPR*, pp. 3686–3696, 2025b.

APPENDIX

A Implementation Details **14**

A.1 E^{2}RPO Configuration 14

A.2 Training Setup 15

B More Results and Discussion **15**

B.1 Reward Design of E^{2}RPO 15

B.2 Reasoning Design in ThinkAxis 15

B.3 Trajectory Supervision vs. Policy Optimization 16

B.4 Why Geometrically Spaced Move Tokens? 16

B.5 Plug-and-Play Improvement across VLMs 17

C Data Curation Details **17**

C.1 Densifying Grounding Supervision 17

C.2 Automated Trajectory Construction 18

D E^{2}RPO Algorithm **19**

E Additional Qualitative Results **21**

E.1 Challenge Cases 21

E.2 Failure Cases 22

E.3 Qualitative Results in Autonomous Driving 23

A IMPLEMENTATION DETAILS

A.1 E^{2}RPO CONFIGURATION

Reward and credit parameters. The initial-localization, final-localization, and process-reward weights are set to $\lambda_0 = 1$, $\lambda_{\text{final}} = 4$, and $\lambda_{\text{proc}} = 4$, respectively. For movement utility, directional correctness and localization progress receive weights $\lambda_{\text{dir}} = 0.6$ and $\lambda_{\text{prog}} = 0.4$. Response advantages are normalized across responses to the same prompt, while route advantages are normalized across all active routes from that prompt's rollout group. Inactive, padded route slots are excluded from the statistics. The normalization constant is $\epsilon = 10^{-6}$.

Counterfactual masking. Mask regions are defined on the merged visual-token grid. Ground-truth boxes are mapped to this grid, and patch membership is determined by cell-center locations. Boundary masks select cells inside the target box whose distance to its nearest edge is at most one merged-grid cell. Routes containing a credited `<add>` instead select target-interior evidence. We use the `sample_union` configuration, combining the selected regions of all active routes in a response into a single counterfactual input. Masked patches are replaced with the image-wise mean patch vector, preserving the visual sequence length and patch positions. The unchanged response is evaluated with the behavior policy under this counterfactual input, without gradients or resampling.

Evidence gate and weight schedule. The evidence score E is the mean action-token log-likelihood drop, measured in nats per token. We use the following bounded gate:

$$g(E) = \begin{cases} 1.00, & E < 0.30, \\ 1.10, & 0.30 \le E < 0.75, \\ 1.25, & E \ge 0.75. \end{cases} \tag{10}$$

Table 8: **Ablation of reward components in E^{2}RPO.** Results are reported on refCOCOg.

Reward Components			refCOCOg	
R^0	R^T	R^{proc}	Acc@0.5 ↑	Acc@0.9 ↑
✔	✔		85.9	55.2
	✔	✔	83.2	**56.4**
✔	✔	✔	**86.4**	56.3

Table 9: **Effect of reasoning formulation.** Localization reasoning improves precision.

Reasoning Form	refCOCOg		ReasonSeg
	Acc@0.5 ↑	Acc@0.9 ↑	gIoU ↑
Action-only	85.2	55.8	62.3
Generic CoT	85.4	55.4	62.5
ThinkAxis	**86.4**	**56.3**	**64.9**

Thus, scores below 0.30 produce no additional evidence credit. The coefficient of the evidence-gated policy-gradient term is linearly increased to 0.1 over the first 100 training steps:

$$\lambda_s = 0.1 \min\left(\frac{s}{100}, 1\right). \tag{11}$$

A.2 Training Setup

Referring expression Benchmarks. For REC benchmarks, cold-start supervision is constructed from a sampled subset of RefCOCO/+/g Mao et al. (2016); Kazemzadeh et al. (2014). Our data curation pipeline converts their box annotations into executable localization trajectories, pairing spatial actions with intermediate reasoning. For reinforcement learning, we use the same $7K$ training data as VisionReasoner Liu et al. (2026; 2025a), covering both single-target and multi-instance grounding.

RealVLG Benchmarks. We construct cold-start trajectories from less than one-third of the official RealVLG training split Li et al. (2026). We then perform reinforcement learning on $9K$ examples sampled from this split, using the same localization-action representation and optimization objectives as in the REC experiments. We evaluate the official evaluation splits using the prescribed metrics.

Autonomous-driving scenarios. We deploy ThinkAxis in real-world autonomous-driving annotation scenarios, covering traffic signs, accident vehicles, road obstacles, and other safety-critical objects. Such applications involve large-scale data and demand high-quality annotations, requiring both complete object coverage and precise localization while avoiding missed, duplicate, or inaccurate predictions. The challenge is further amplified in wide-view driving images, where targets can be small, distant, densely distributed, and visually ambiguous. We focus on traffic-sign localization as a representative challenging setting, where these difficulties frequently occur together and place particularly high demands on localization quality. We use an internal traffic-sign development set comprising 2,805 images from 70 driving scenes, with 11,131 annotated objects. For a controlled comparison, we train a GRPO baseline using the same data and model backbone as ThinkAxis.

B More Results and Discussion

B.1 Reward Design of E^{2}RPO

We further analyze how different rewards shape the localization trajectory. The terminal reward R^T supervises the final localization quality, while the process reward R^{proc} encourages effective corrections along the trajectory. As shown in Tab. 8, adding R^{proc} substantially improves strict localization accuracy, confirming its role in promoting precise coordinate correction. However, without supervising the initial prediction, optimizing process improvement alone can encourage the model to start from a poorer localization so that subsequent actions obtain larger gains, leading to a clear drop in $\text{Acc@}0.5$. The initial reward R^0 therefore anchors the starting localization and prevents such reward-seeking behavior. Jointly supervising the initial, intermediate, and final localization yields the most balanced performance, ensuring that process-level improvement does not come at the expense of localization quality.

B.2 Reasoning Design in ThinkAxis

Standard grounding annotations typically provide only the final target location, offering little supervision on *what is wrong* with an intermediate localization and *how it should be corrected*. To fill this

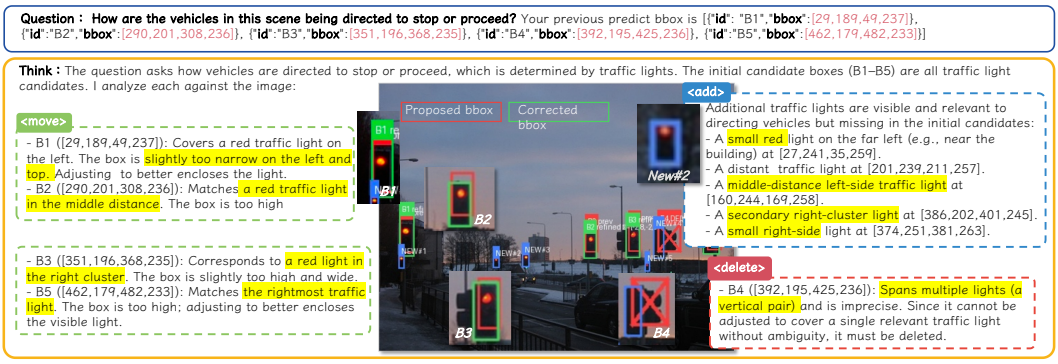

Figure 8: **Example of a localization correction sample constructed by our data pipeline.** Each sample explicitly reasons about localization errors and the required corrections, paired with executable actions and the final precise localization result.

Table 10: **Trajectory data contribution.** Comparison with policy optimization.

Training Stage	refCOCOg	
	Acc@0.5 ↑	Acc@0.9 ↑
Base	84.1	52.0
+ Trajectory SFT	85.3	53.6
+ E^{2}RPO	**86.4**	**56.3**

Table 11: **Effect of localization representation on RefCOCOg.** Relative actions improve correction efficiency and localization precision.

Localization Representation	Avg. Steps ↓	Acc@0.9 ↑
Absolute coordinates	4.2	52.7
Raw pixel offsets	3.4	54.4
Geometric move tokens	**2.2**	**56.3**

gap, our data pipeline automatically constructs correction-oriented reasoning traces that diagnose localization errors, determine the required adjustment, and associate each decision with executable actions such as `<move>`, `<add>`, and `<delete>`. As illustrated in Fig. 8, this supervision teaches the model not only which localization action to take, but also why the current prediction is inaccurate and how it should change.

We compare different reasoning formulations to understand the role of correction-oriented reasoning. As shown in Tab. 9, Action-only directly predicts localization actions without reasoning about the underlying error, while Generic CoT provides general semantic reasoning but does not explicitly determine how the current localization should be corrected. Generic CoT therefore brings only marginal gains over Action-only and even slightly decreases Acc@0.9. In contrast, ThinkAxis explicitly connects error diagnosis with the subsequent localization action, improving Acc@0.5 to **86.4**, Acc@0.9 to **56.3**, and ReasonSeg gIoU to **64.9**. These results show that effective localization reasoning requires more than generic thinking: the reasoning trace should explicitly identify localization errors and guide how they are corrected.

B.3 Trajectory Supervision vs. Policy Optimization

We disentangle the roles of trajectory supervision and subsequent policy optimization in Tab. 10. Trajectory SFT first teaches the model the structure of ThinkAxis reasoning, including how localization thoughts and executable actions form a coherent trajectory. This stage improves Acc@0.5 from 84.1 to 85.3 and Acc@0.9 from 52.0 to 53.6. E^{2}RPO further improves them to **86.4** and **56.3**, with a larger gain under the stricter criterion. These results disentangle the contributions of data and optimization: the constructed trajectory data establishes the localization reasoning structure, while policy optimization further improves the effectiveness and precision of localization decisions.

B.4 Why Geometrically Spaced Move Tokens?

As shown in Tab. 11, replacing absolute-coordinate regeneration with relative offsets already shortens the correction trajectory and improves precise localization, supporting the benefit of explicitly representing localization changes. However, raw offsets still permit arbitrarily small updates that contribute little geometric progress. Geometrically spaced move tokens further reduce the average

Table 12: **Cross-backbone generalization of ThinkAxis.** ThinkAxis improves localization predictions from different VLM backbones without retraining them.

Backbones	refCOCO	refCOCO+	refCOCOg	Avg.
	Test-A	Test-A	Test	
Qwen2-VL-7B Wang et al. (2024)	92.4	90.2	88.0	90.2
+ ThinkAxis	94.3	92.1	90.2	92.2
InternVL3-8B Zhu et al. (2025a)	93.1	91.1	89.4	91.2
+ ThinkAxis	94.5	92.4	90.8	92.6

number of steps while achieving the highest $\mathrm{Acc@0.9}$, indicating that a compact set of meaningful movement scales improves both correction efficiency and boundary precision.

We next explain why geometric spacing provides this favorable behavior. Let the positive displacement levels be $q_k = \epsilon\beta^{k-1}$, $k = 1, \ldots, K$. Under a fixed token budget, the maximum representable displacement grows exponentially with K, while the minimum resolution remains ϵ. The same vocabulary can therefore cover large localization errors while retaining fine-grained movements near the target. More importantly, geometric spacing provides a bounded relative quantization error. For a normalized boundary residual r_t within the range covered by the vocabulary, we select the largest same-direction displacement q_t that does not overshoot the target. It therefore satisfies $q_t \leq |r_t| < \beta q_t$. Hence, the selected movement accounts for more than a $1/\beta$ fraction of the remaining residual, giving

$$|d_{t+1}| < \left(1 - \frac{1}{\beta}\right)|d_t|. \tag{12}$$

With $\beta = 2$, each valid correction removes more than half of the remaining boundary error. Geometric spacing thus combines a wide correction range with bounded relative quantization error, explaining why a compact movement vocabulary can support both rapid localization updates and precise boundary adjustment.

B.5 Plug-and-Play Improvement across VLMs

Beyond end-to-end training, we examine whether ThinkAxis can improve localization predictions from other VLMs. Given an off-the-shelf backbone prediction as the initial localization, ThinkAxis performs subsequent localization correction without retraining the backbone. As shown in Tab. 12, ThinkAxis improves both Qwen2-VL-7B Wang et al. (2024) and InternVL3-8B Zhu et al. (2025a), increasing the average accuracy from 90.2 to 92.2 and from 91.2 to 92.6, respectively. These results suggest that ThinkAxis can transfer across the tested backbones and may serve as a reusable localization corrector without backbone retraining.

From a deployment perspective, such transferability implies that ThinkAxis can be integrated as an external refinement module to improve localization reliability without necessitating modifications to the primary backbone. More broadly, these findings support a modular generator-refiner architecture, where a high-capacity backbone generates initial proposals and ThinkAxis provides a specialized geometric refinement stage. This modularity is particularly advantageous in scenarios where the computational cost of updating or retraining the primary backbone is prohibitive, as the refinement component can be efficiently reused across diverse model families.

C Data Curation Details

C.1 Densifying Grounding Supervision

Conventional grounding data provide sparse supervision: each sample specifies the target location but leaves the process of reaching and correcting it unobserved. Our construction unfolds a single annotation into intermediate localization hypotheses, executable actions, and action-aligned reasoning. This turns final-only supervision into trajectory-level supervision over localization errors and the decisions needed to correct them, increasing the information density of existing data without additional manual annotation.

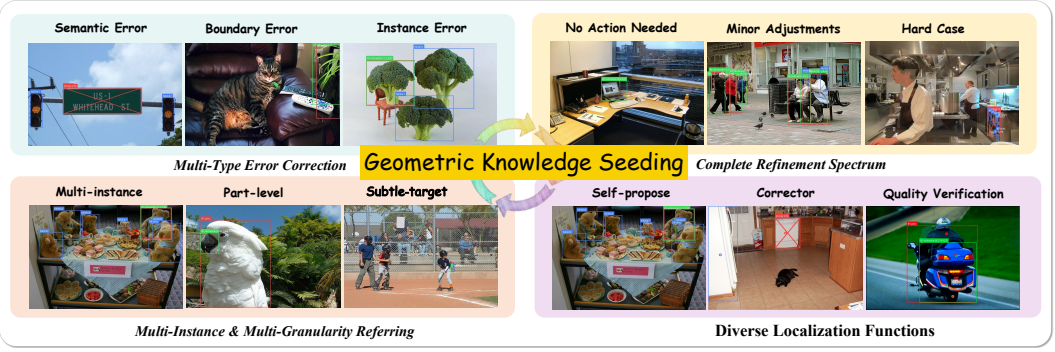

Figure 9: **Richer supervision mined from grounding data.** Starting from standard target-box annotations, trajectory construction extracts substantially richer process supervision for localization reasoning.

As shown in Fig. 9, model-generated starting predictions further expose diverse localization failures beyond simple boundary errors. Executable trajectories convert these naturally occurring failures into learnable correction and recovery behaviors, covering the localization process from hypothesis formation and revision to determining when the prediction is sufficiently accurate.

C.2 Automated Trajectory Construction

We automatically convert standard grounding annotations into executable localization reasoning trajectories through the following steps.

Step 1: Model-induced initialization. Rather than perturbing ground-truth boxes, we use predictions from a frozen grounding model as the initial localization hypotheses, exposing structured errors closer to those encountered at inference time. For multi-instance expressions, predictions and targets are aligned by maximum-IoU one-to-one matching. Matched predictions are retained or corrected according to their localization quality, unmatched predictions are removed with `<delete>`, and missing targets are recovered with `<add>`.

Step 2: Geometry-driven action planning. The complete action trajectory is determined before any textual reasoning is generated. Predictions with IoU above 0.95 terminate with `<end>`, those in $[0.5, 0.95)$ are corrected with `<move>`, and predictions below 0.3 or unsuitable for local correction are replaced through `<delete>`→`<add>`→`<move>`. Cases in $[0.3, 0.5)$ are retained only when an additional visual check confirms that the prediction still refers to the correct target. For each valid localization, we select the largest same-direction movement token that does not overshoot the corresponding target boundary. Movements below three pixels are mapped to `<mv0>`. Each action is executed before the next is planned, and planning stops once IoU reaches 0.95, no effective movement remains, or four `<move>` actions have been applied. For `<add>` trajectories, directly inserting the ground-truth box would make the subsequent reasoning trivial. We instead initialize the added instance with a coarse target-associated box whose IoU is sampled from $[0.30, 0.50)$, $[0.50, 0.70)$, or $[0.70, 0.85)$ with probabilities 0.15, 0.50, and 0.35, respectively, and then apply the same movement planner.

Step 3: Action-conditioned reasoning annotation. Once the geometric trajectory is fixed, a multimodal teacher generates a concise `<think>` span for each transition using the image, referring expression, current localization, and prescribed action. The teacher explains the visual evidence supporting the decision but cannot alter the action type, movement tokens, resulting box, or trajectory length. Ground-truth coordinates, IoU values, pixel offsets, movement-token values, and other private supervision are prohibited from the final reasoning text.

Step 4: Replay verification and balancing. All trajectories are replayed using the same action semantics as ThinkAxis. We discard malformed or truncated sequences, invalid actions, unresolved ambiguous cases, low-quality rationales, and samples containing private-supervision leakage. Both step-level and trajectory-level annotation quality must reach at least four on a five-point scale. Because raw predictions are dominated by already accurate localizations and short `<end>`-only trajectories, we rebalance the retained data across localization difficulty and action types. The cold-

start mixture contains 10% severe errors with initial IoU below 0.5, 30% corrections in $[0.5, 0.75)$, 20% in $[0.75, 0.9)$, 10% fine boundary corrections in $[0.9, 0.95)$, 15% terminal or near-terminal cases, and 15% multi-instance or `<add>`/`<delete>` trajectories.

[Task and input]

You are writing supervision text for a visual-action grounding model. Your job is not to describe the JSON fields. Your job is to write the short visual thought that should appear immediately before each action during training.

The image has colored proposal boxes labeled P1, P2, ... The query can refer to zero, one, or multiple objects. In one API response, write both:
1. proposal_thinks: the visual search thought for each initial proposal label.
2. steps: final training thoughts and quality scores for each planned edit action.

If there are no initial proposal boxes, proposal_thinks must be an empty list. This is not an error. Judge the trajectory by whether the add/move/end steps locate the referred target(s) using visible evidence.

[Important rules]

- Do not mention ground truth, GT, hidden/private boxes, IoU, numeric scores, token values, or coordinates.
- Do not copy coordinates into the description.
- The proposal think must sound like the model is searching the image from the query, not commenting on any rectangle.
- In proposal think text, do not mention the colored rectangle itself. Avoid words like bounding box, proposal, highlighted, candidate, label, or annotation. The word "box" is allowed only when it means a real physical object in the image, such as a donut box, bento box, or cardboard box.
- In proposal think text, do not write "captures", "contains", or "highlights" if the subject is an artificial rectangle/proposal.
- Write proposal_thinks as if no colored rectangles exist; focus only on visual evidence and target identity.
- The planned_actions list is only a skeleton: action type, box id, rough location, and edge movement.
- steps[].description is the final text that will be placed inside <think> before the action token.
- Write each steps[].description from scratch using the image, query, action type, rough_area_hint, and edge_shift_hint.
- Do not begin steps[].description with "Add N1:", "Move P1:", "Stop editing", or similar mechanical labels.
- Do not say "the box" unless the object is a real physical box in the image.
- Do not talk like a reviewer. The description should sound like the model's own localization thought.
- For delete, explain why the current target hypothesis is visually inconsistent with the query.
- For add, explain which missing referred instance should be added, using visual cues from the query.
- For move, explain what boundary or object-part mismatch is being corrected.
- For <end>, explain why this box/set should stop changing.
- In multi-target cases, descriptions must distinguish instances when possible, e.g. left/right, color, pose, count, or relation.
- Keep descriptions concise but specific, usually 12-28 words.
- Avoid generic descriptions such as "create a coarse box for a missing referred instance", "refine toward the matched referred instance", "converged", or "does not match any referred target".
- If the input planned step has a generic placeholder, you must replace it with a specific visual reason. Copying a generic placeholder should be treated as low quality.

[Proposal-think style examples]

- BAD: "The box captures the lower banana."
- GOOD: "I look for the lower banana, separated from the upper banana by its lower position and curved yellow body."
- BAD: "The highlighted region is a person in the background."
- GOOD: "I check the background person, but the query asks for the woman on the right in the red shirt."
- BAD: "P1 is already correct."
- GOOD: "The pink frosted donut in front is identifiable by its white sprinkles and its position nearest the viewer."

[Step-description style examples]

- BAD move: "Move P1: refine the box toward the matched referred instance."
- GOOD move: "Shift the left edge inward to remove the neighboring donut while keeping the full pink frosting visible."
- GOOD move: "Raise the top boundary toward the giraffe's ears and neck, while trimming extra sky from the right side."
- BAD end: "Stop editing P1: converged."
- GOOD end: "Stop editing P1 because it now covers the middle elephant without including the calf or the larger adult beside it."
- GOOD end: "The left giraffe is separated from the nearby giraffe and its full neck and legs remain inside."
- BAD delete: "Delete P2: it does not match any referred target instance."
- GOOD delete: "Delete P2 because it points to background foliage, while the query asks for the bush on the far right."
- GOOD delete: "This hypothesis follows the rider, but the referred woman is seated behind him with red hair."
- BAD add: "Add N1: create a coarse box for a missing referred instance."
- GOOD add: "Look near the far-left edge for the barely visible man in a dark sleeve, since no initial target covers him."
- GOOD add: "Add the rightmost duck standing apart from the other birds, using its orange bill and position near the waterline."
- GOOD move: "Pull the bottom edge down to include the chef's blue apron while keeping the stove controls outside the target."
- GOOD end: "The pink slice is now isolated from the whole pizza and the white sign beside it, so no more edits are needed."

[Output]

Return strict JSON:

```
{
  "proposal_thinks": [
    {"box_id": "P1", "think": "visual search thought for this initial proposal"}
  ],
  "trajectory_quality_score": 0-5,
  "trajectory_quality_reason": "short reason",
  "steps": [
    {"step": 0, "description": "...", "quality_score": 0-5, "quality_reason": "..."}
  ]
}
```

[Completeness and self-check]

You must return exactly one proposal_think for each proposal box id and exactly one step annotation for each planned step id. When proposal_boxes is empty, return "proposal_thinks": [] and still score the trajectory normally. For each step item, output only step, description, quality_score, and quality_reason. Do not echo action names, roles, boxes, coordinates, or the planned trajectory fields. Do not wrap the JSON in markdown code fences. Before finalizing, silently check every steps[].description:
1. It must mention concrete visual evidence from the image/query.
2. It must not be a generic placeholder.
3. It must be usable inside <think> without editing.

[Planned trajectory]

{PLANNED_TRAJECTORY_JSON}

Figure 10: **Prompt template for data trajectory synthesis.** The template guides the teacher model to generate diagnostic rationales coupled with executable updates for spatial alignment.

D E^{2}RPO ALGORITHM

Algorithm 1 summarizes the complete training procedure. E^{2}RPO separates learning the quality of a complete response from assigning additional credit to individual spatial decisions. Response-level

Algorithm 1: E^{2}RPO: Edge-Evidence Relative Policy Optimization

Input: Dataset $\mathcal{D} = \{(I, q, Y^\star)\}$; policy π_θ; frozen reference π_{ref}; batch size B; group size N; clipping parameter ϵ_c; normalization constant $\epsilon > 0$; KL coefficient β; evidence-weight schedule λ_s; bounded gate g.

Output: Optimized policy π_θ.

Define $\text{Norm}(v; \mathcal{V}) = (v - \text{mean}(\mathcal{V}))/(\text{std}(\mathcal{V}) + \epsilon)$

for *each training iteration* s **do**

- Freeze behavior-policy snapshot $\theta_{\text{old}} \leftarrow \theta$
- Sample $\{(I_i, q_i, Y_i^\star)\}_{i=1}^{B} \sim \mathcal{D}$
- **for** $i = 1, \ldots, B$ **do**
  - // 1. Sample trajectories and evaluate geometric utility
  - **for** $j = 1, \ldots, N$ **do**
    - Sample $y_{i,j} \sim \pi_{\theta_{\text{old}}}(\cdot \mid I_i, q_i)$ and execute its spatial actions
    - Compute response reward $R_{i,j}$
    - Match hypotheses to $Y_i^\star$ and extract active routes $\mathcal{R}_{i,j}$
    - Obtain utility $U_{i,j,r}$ and credited token set $\mathcal{T}_{i,j,r}$ for each $r \in \mathcal{R}_{i,j}$, assigning each credited token to at most one route
    - Cache $\ell_{i,j,t}^{\text{orig}} \leftarrow \log \pi_{\theta_{\text{old}}}(y_{i,j,t} \mid I_i, q_i, y_{i,j,<t})$ for all valid response tokens
  - // 2. Normalize response and route credit separately
  - $\mathcal{V}_i^{\text{resp}} \leftarrow [R_{i,j}]_{j=1}^{N}$
  - $\mathcal{V}_i^{\text{route}} \leftarrow [U_{i,j,r}]_{j=1,\ldots,N;\, r \in \mathcal{R}_{i,j}}$
  - **for** $j = 1, \ldots, N$ **do**
    - $A_{i,j}^{\text{resp}} \leftarrow \text{Norm}(R_{i,j}; \mathcal{V}_i^{\text{resp}})$
    - **foreach** $r \in \mathcal{R}_{i,j}$ **do**
      - $A_{i,j,r}^{\text{route}} \leftarrow \text{Norm}(U_{i,j,r}; \mathcal{V}_i^{\text{route}})$
      - $\widetilde{A}_{i,j,r} \leftarrow 0$
    - // 3. Construct one shared counterfactual per response
    - $\mathcal{R}_{i,j}^{\text{valid}} \leftarrow \{r \in \mathcal{R}_{i,j} : r \text{ has a valid matched GT box, } |\mathcal{T}_{i,j,r}| > 0\}$
    - **foreach** $r \in \mathcal{R}_{i,j}^{\text{valid}}$ **do**
      - Select patches $\mathcal{M}_{i,j,r}$ from the matched GT box: interior if the route contains a credited `<add>`, boundary band otherwise
    - $\mathcal{M}_{i,j} \leftarrow \bigcup_{r \in \mathcal{R}_{i,j}^{\text{valid}}} \mathcal{M}_{i,j,r}$
    - **if** $\mathcal{M}_{i,j} \neq \varnothing$ **then**
      - $\widetilde{I}_{i,j} \leftarrow \text{MeanFill}(I_i, \mathcal{M}_{i,j})$
      - Score the unchanged sequence without gradients: $\ell_{i,j,t}^{\text{mask}} \leftarrow \log \pi_{\theta_{\text{old}}}(y_{i,j,t} \mid \widetilde{I}_{i,j}, q_i, y_{i,j,<t})$
      - // 4. Compute an evidence bonus for positive routes
      - **foreach** $r \in \mathcal{R}_{i,j}^{\text{valid}}$ **do**
        - $E_{i,j,r} \leftarrow \dfrac{1}{|\mathcal{T}_{i,j,r}|} \displaystyle\sum_{t \in \mathcal{T}_{i,j,r}} \left(\ell_{i,j,t}^{\text{orig}} - \ell_{i,j,t}^{\text{mask}}\right)$
        - $\widetilde{A}_{i,j,r} \leftarrow [A_{i,j,r}^{\text{route}}]_+ \mathbf{1}[U_{i,j,r} > 0]\, [g(E_{i,j,r}) - 1]_+$
    - // 5. Replace base action credit; assign the extra bonus
    - Initialize $A_{i,j,t}^{\text{base}} \leftarrow A_{i,j}^{\text{resp}}$ for all valid response tokens
    - **foreach** $r \in \mathcal{R}_{i,j}$ **do**
      - Replace $A_{i,j,t}^{\text{base}} \leftarrow A_{i,j,r}^{\text{route}}$ for $t \in \mathcal{T}_{i,j,r}$
      - Assign $\widetilde{A}_{i,j,r}$ to these token positions in $\mathcal{J}_{\text{edge}}$
    - Detach all advantages and evidence scores
  - // 6. Update on original inputs with separate clipped objectives
  - Compute $\mathcal{J}_{\text{base}}$ using $A_{i,j,t}^{\text{base}}$ and $\mathcal{J}_{\text{edge}}$ using $\widetilde{A}_{i,j,r}$, each with clipping parameter ϵ_c
  - Update θ by maximizing $\mathcal{J}_{\text{base}} + \lambda_s \mathcal{J}_{\text{edge}} - \beta \mathcal{L}_{\text{KL}}$

advantages supervise all valid output tokens, preserving a learning signal for intermediate reasoning and overall task completion, while route-level credit selectively reinforces the corresponding action tokens.

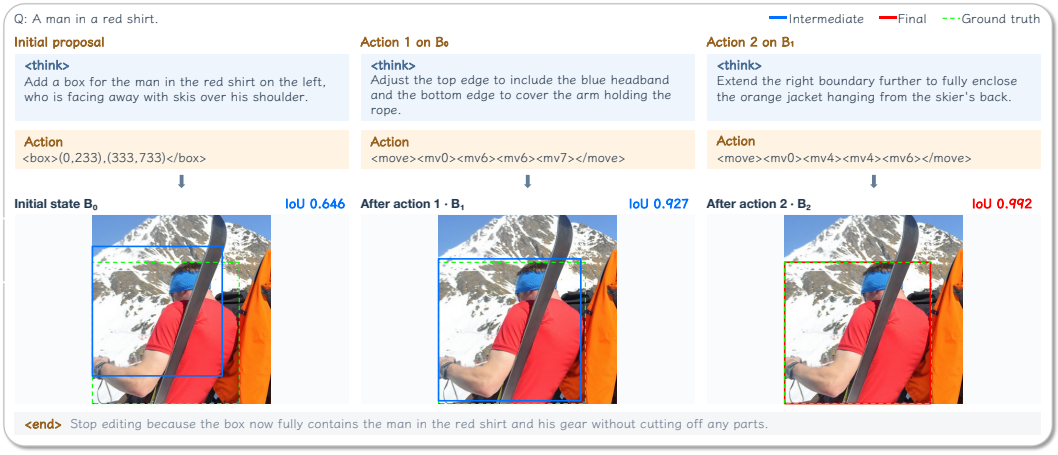

Figure 11: **Precise localization after semantic resolution.** Once the target is correctly identified, successive localization actions correct its spatial extent, increasing IoU from 0.646 to 0.992. Each displayed localization follows execution of the action directly above it.

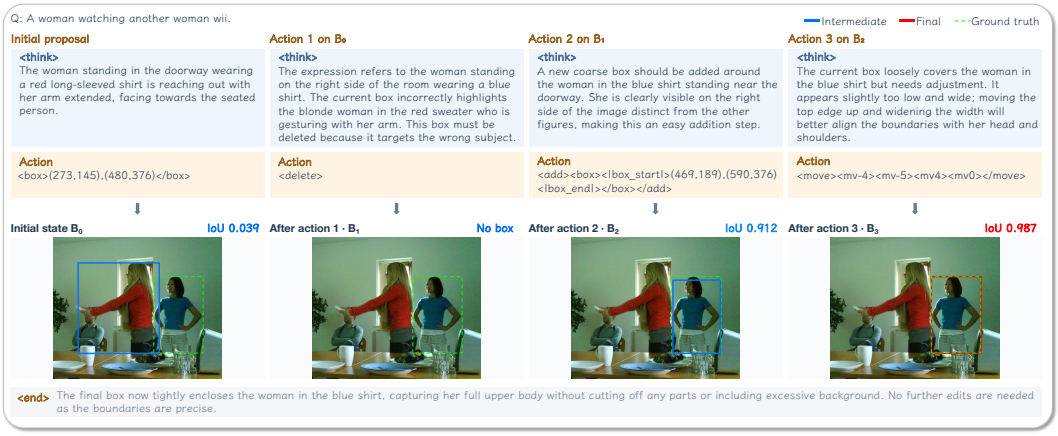

Figure 12: **Localization guided by semantic reassessment.** Semantic reasoning replaces an incorrect target hypothesis, after which localization actions recover precise boundaries. Each displayed localization follows execution of the action directly above it.

E Additional Qualitative Results

E.1 Challenge Cases

Ambiguous spatial extent in a cluttered scene. In Fig. 11, recognizing the skier by his red shirt does not readily determine the full target extent. The skis, carried equipment, and nearby objects create competing visual boundaries, making it difficult to distinguish the target's spatial extent from its surroundings. The initial proposal captures the correct person but excludes part of the annotated region. Two successive corrections increase IoU from 0.646 to 0.927 and then 0.992, illustrating how explicit spatial updates can resolve residual boundary errors after target identity is established.

Relational ambiguity with a salient distractor. Fig. 12 involves two women whose roles must be distinguished through the query's relational description. The initial proposal selects the gesturing woman in red rather than the woman watching her, confusing the participant performing the salient action with the intended referent. Refining this proposal locally would leave the identity error unresolved. ThinkAxis instead deletes the incorrect hypothesis and adds a proposal for the woman in blue, reaching 0.912 IoU before further refinement raises it to 0.987. The key challenge is therefore recognizing when to abandon an existing hypothesis, rather than continuing to refine its boundaries.

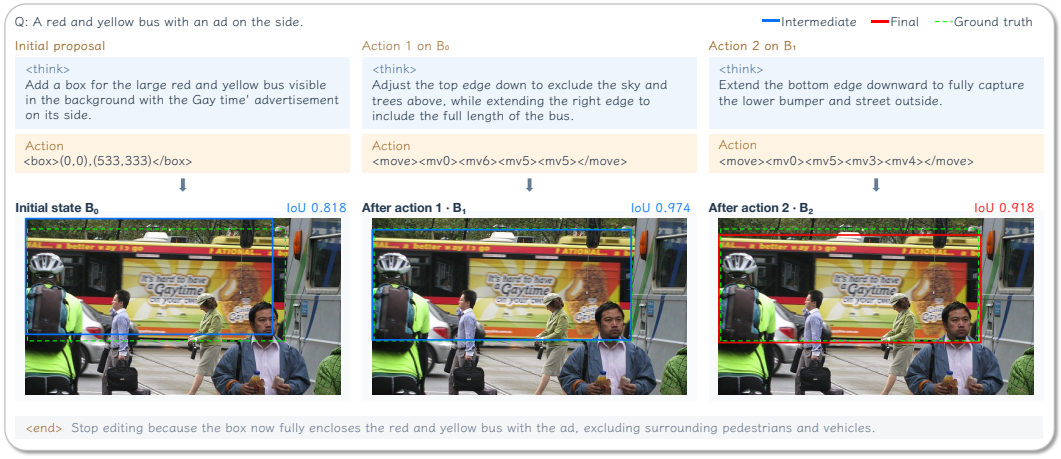

Figure 13: **Overcorrection after an accurate localization.** An unnecessary second update reduces IoU from 0.974 to 0.918, illustrating the importance of timely termination. Each displayed state follows execution of the action directly above it.

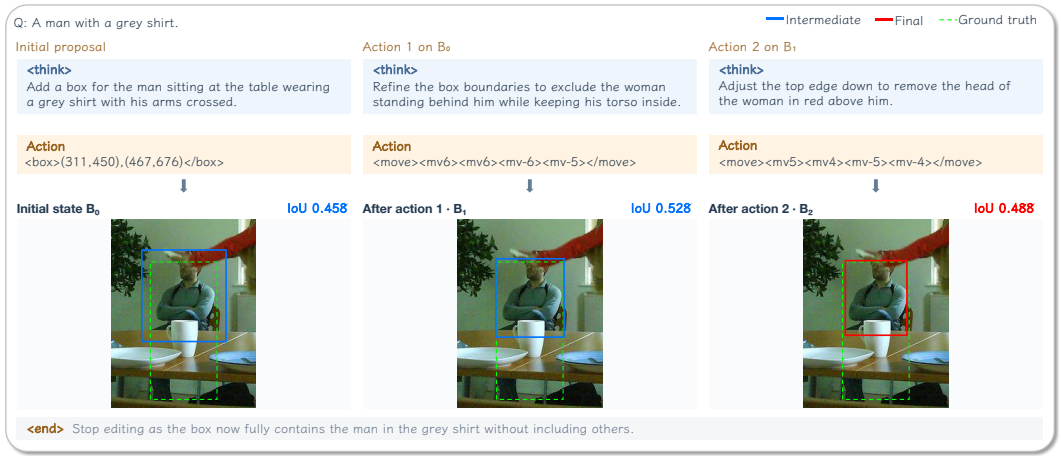

Figure 14: **Correct target identification with incomplete coverage.** Successive contractions retain the target's upper body but fail to recover its full annotated extent, despite a confident stop decision. Each displayed state follows execution of the action directly above it.

E.2 Failure Cases

Overcorrection near an accurate solution. In Fig. 13, the first correction improves the bus localization from 0.818 to 0.974 IoU. A subsequent update slightly decreases IoU to 0.918, as the intended boundary extension also affects other box edges. This failure reflects the challenge of making fine-grained geometric decisions near an accurate solution, where even a small unnecessary update can introduce regression.

Correct identity but incomplete spatial coverage. Fig. 14 shows a case where the model identifies the correct person but predicts only the upper body rather than the full annotated extent. The first update improves IoU from 0.458 to 0.528, while further adjustment decreases it to 0.488. The remaining error arises from a mismatch between semantic recognition and annotation-level geometric completeness: recognizing the target does not necessarily guarantee recovering its full spatial extent.

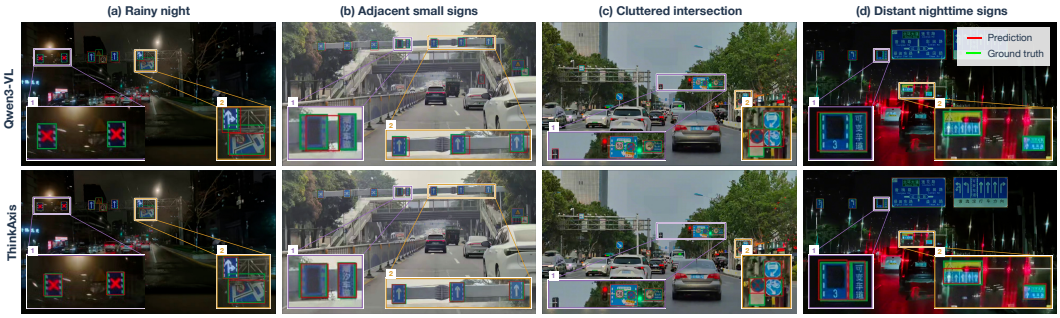

Figure 15: **Localization in challenging autonomous-driving scenes.** Compared with Qwen3-VL, ThinkAxis recovers missed signs and improves boundary alignment across rainy nights, adjacent small targets, cluttered intersections, and distant nighttime objects. Paired enlargements use identical crop regions and magnification.

E.3 Qualitative Results in Autonomous Driving

Fig. 15 presents qualitative comparisons on challenging traffic-sign localization in driving scenes. These scenarios contain small and distant objects, dense arrangements, and large appearance variations caused by illumination and weather conditions. Compared with Qwen3-VL, ThinkAxis recovers missed signs and produces localization results with more complete coverage and better boundary alignment. The enlarged regions highlight two complementary improvements: recovering overlooked targets in rainy and nighttime scenes, and separating adjacent signs with more precise spatial boundaries. These examples illustrate the benefit of localization evolution in complex driving scenarios, where a single prediction may struggle to capture all relevant objects and their precise spatial extent. Remaining errors mainly occur in highly cluttered regions with severe visual ambiguity.