STAYING CONSISTENT, STAYING ON TARGET: REVISABLE REFERENCES FOR VISUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual references enable vision-language models to reuse object-level information across successive questions. However, when an early localization is incorrect, reusing it can propagate the mistake through later reasoning, even when the resulting answers remain mutually consistent. Recovering from such errors requires revising the localization while preserving the object intended by the task. We address this requirement through revisable references, which separate task-defining information from the model's localization estimates. In a binary repair model, we characterize when rewards for historical agreement oppose task-beneficial corrections. This motivates Referent-Preserving Policy Optimization ($\mathrm{RP}^2\mathrm{O}$), a reinforcement learning framework that learns reference revisions from their consequences for subsequent reasoning. $\mathrm{RP}^2\mathrm{O}$ jointly optimizes revision and execution: edits are assessed against retaining the original reference under the same executor, while continuations are compared within the same edited state. These comparisons connect both decisions to downstream task utility. Answer accuracy alone, however, does not establish whether the maintained reference has been repaired. We therefore design a controlled evaluation protocol that separately assesses reference recovery, preservation of the intended object, and answer correctness, using paired reference semantics and shared starting states under matched information, supervision, and computational budgets.

1 INTRODUCTION

Multi-turn visual reasoning requires a model to carry visual references across a sequence of dependent questions. Advances in multi-round segmentation, reasoning-integrated perception, and explicit region citation make localized regions an increasingly important part of the reasoning context (Wang et al., 2025; Liu et al., 2026b; Sun et al., 2026). When an early localization is wrong, later answers may remain consistent with the stored region while referring to the wrong object (Figure 1). Agreement across turns can therefore coexist with a persistent grounding error. The intended object is specified by the task description or an explicit selection, while the cached box records the model's current estimate of its location. Reliable reference maintenance must allow this estimate to change while preserving the intended object.

This motivates our central question: how can a model learn to revise visual references while preserving the object under discussion? Self-correction research shows that improving initial-answer accuracy need not yield effective recovery (Kumar et al., 2025). For persistent visual references, rewarding agreement with an earlier region can favor keeping an error even when a revision would improve subsequent answers. We characterize this conflict in a binary repair model, identifying when historical citation incentives reverse the update toward a task-beneficial repair.

We propose Referent-Preserving Policy Optimization ($\mathrm{RP}^2\mathrm{O}$), a reinforcement learning framework that jointly optimizes reference revision and subsequent reasoning. The framework holds task-defining information fixed while learning to update cached localizations. Since a revised reference will be reused by later questions, we assess each edit through the utility of subsequent answers. Its value thus depends on both the questions that follow and the model reasoning from the revised reference.

Learning from these answers requires distinguishing the value of an edit from the quality of reasoning under that edit. Revision credit compares an edit with retaining the original reference under the same

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.

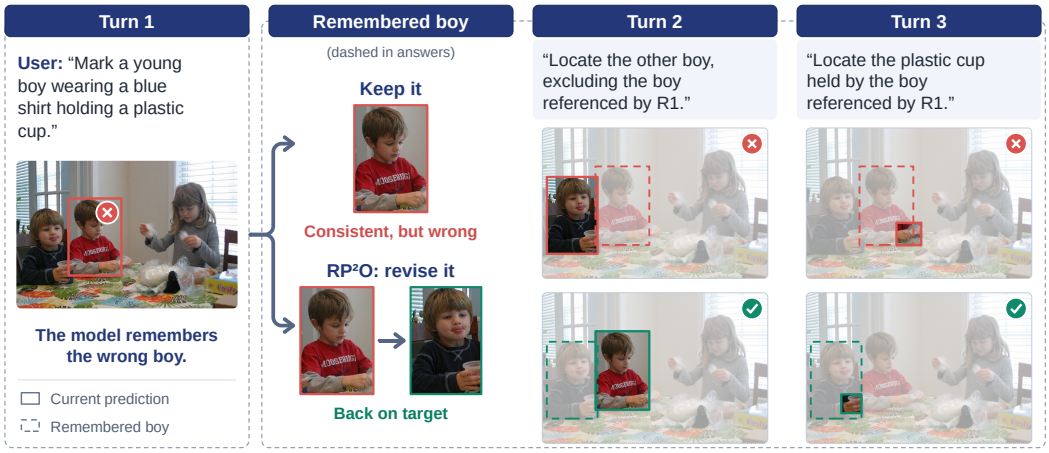


Figure 1: **Consistent is not correct.** The user asks for the boy in a blue shirt holding a plastic cup, but the model remembers the boy in red. If that reference is kept (top; the No-edit condition), later answers are resolved from it (dashed box): they agree with the remembered boy, yet concern the wrong one. $\mathrm{RP}^2\mathrm{O}$ (bottom) revises only the remembered box, keeping the user's description, so the same questions are answered for the intended boy.

executor; execution credit compares continuations from the same edited state. These signals jointly train a shared vision-language model. Under explicit sampling assumptions, we derive an unbiased estimator of the joint policy gradient at the rollout parameters. Deployment requires one revision decision followed by one continuation.

Evaluating this capability requires separating fewer initial errors from better recovery, and answer success from reference recovery. Grounding shortcuts and oracle selection can obscure these distinctions (Dong et al., 2026; Tripathy & Krishnan, 2026). We introduce a controlled protocol with shared starting states to compare recovery from the same errors, semantic-role pairs to test preservation of the intended object, and joint reference–answer assessment to check whether both are correct. Natural trajectories, standard grounding, and external dialogue evaluations complement these diagnostics. Controlled comparisons share observable input fields and annotation access and account for generated rollout tokens when comparing learned revision, utility choices, and credit assignment. On Qwen2.5-VL-7B, $\mathrm{RP}^2\mathrm{O}$ improves joint recovery from 66.05% to 72.58% over answer-reward GRPO, while keeping endpoint answer accuracy within one percentage point of the baseline. Further analyses show that citation rewards can suppress beneficial reference revisions, consistent with the incentive conflict identified by our analysis.

We formalize referent-preserving revision and characterize when historical agreement incentives oppose task-beneficial repair in a binary model. We propose $\mathrm{RP}^2\mathrm{O}$ to jointly learn reference revision and subsequent reasoning from downstream task utility. We introduce a controlled evaluation protocol that assesses reference recovery, preservation of the intended referent, and answer correctness under shared information and supervision.

2 Related Work

Grounded reasoning with persistent references. Multi-round segmentation and reasoning-integrated perception support reasoning with localized visual evidence (Wang et al., 2025; Liu et al., 2026b). Grounded operations and perceptual or counterfactual training further strengthen the use of visual evidence (Sarch et al., 2025; Huang et al., 2026b; Wang et al., 2026; Xu et al., 2026; Huang et al., 2026a). For references carried across turns, RegionReasoner combines explicit region citation with global–local consistency to support reuse and refinement (Sun et al., 2026). Our focus is how refinement preserves the intended object when a stored localization is wrong. We separate the information defining that object from the localization that can be revised.

Self-correction and state revision. Self-correction can act on successive responses, as in the reinforcement learning approaches SCoRe and calibrated multi-attempt training (Kumar et al., 2025; Ildiz et al., 2026). Visual recovery methods learn from reasoning errors or revisit perceptual evidence: VRRL trains from erroneous prefixes, ReGround selectively re-examines visual input, and ChartVSR refines pixel anchors before structured decoding (Tang et al., 2026; Peng et al., 2026; Li et al., 2026). Intermediate state updates can also be learned from task outcomes: VTool-R1 trains visual tool use from final rewards, while Memory-R1 learns memory operations and answer generation from downstream question-answering rewards (Wu et al., 2026; Yan et al., 2026). We study visual references inherited by later questions, with a fixed task referent and joint learning of reference revision and subsequent reasoning.

Credit assignment in policy optimization. Policy optimization methods use different comparisons to assign credit. GRPO centers rewards within sampled groups, whereas ReMax uses a response-based baseline (Shao et al., 2024; Li et al., 2024). For multi-turn decisions, REFUEL uses differences in future returns from a shared conversation prefix as regression targets (Gao et al., 2025). Local or nested comparisons in GiGPO and TreeRPO, and rank-level advantages in Rank-GRPO, further adapt credit to the decision structure (Feng et al., 2025; Yang et al., 2025; Zhu et al., 2026). We compare edits against retaining the original reference under the same executor, and continuations against alternatives from the same edited state.

Diagnostic evaluation of recovery. Multi-turn degradation and error propagation motivate tracking performance across turns (Laban et al., 2026; Liu et al., 2026a; Jiang et al., 2026). RegionDial-Bench evaluates grounding across dependent turns (Sun et al., 2026), while interactive grounding benchmarks examine clarification and repair (Wang & Rambow, 2026). Diagnostic evaluations address how apparent gains arise: Ref-Adv targets grounding shortcuts, while IVT distinguishes submitted predictions from oracle-selected intermediate results (Dong et al., 2026; Tripathy & Krishnan, 2026). Recovery analyses also track damage to initially correct predictions (Kumar et al., 2025; Peng et al., 2026). For persistent references, we separately assess recovery from shared starting states and correctness of both the maintained reference and the answer. Semantic-role pairs additionally test whether revision preserves the task's intended object.

3 METHODOLOGY

Referent-Preserving Policy Optimization ($\mathrm{RP}^2\mathrm{O}$) jointly trains reference revision and subsequent reasoning. It holds task-defining information fixed while allowing cached localizations to change. Downstream task utility evaluates the resulting continuations, with separate credit for the revision decision and execution under the edited state. At inference time, a shared visual language model makes one revision decision and continues from the resulting state.

3.1 REVISABLE REFERENCES

We represent a visual reference by its task-defining information and a revisable localization. At a revision opportunity, the observable state is $x = (I, h, r)$, comprising image I, available history and current query h, and active reference $r = (d, s, b, z)$. The record contains an originating description d, optional defining selection s, cached box b, and semantic role $z \in \{\mathsf{E}, \mathsf{D}\}$. The intended object is

$$o^\star(x) := \begin{cases} \mathrm{Resolve}(I, h, d), & z = \mathsf{E} \quad \text{(estimate)}, \\ \mathrm{Select}(I, s), & z = \mathsf{D} \quad \text{(definition)}. \end{cases} \tag{1}$$

The operators specify task semantics; the policy receives the observable fields, with identical access for all compared methods. The semantic role specifies how the intended object is established; it does not indicate whether the cached localization is correct. We consider unique, reference-dependent targets with sufficient originating information; Appendix A states the admissibility conditions and discusses multiple references.

Let $\mathcal{U}$ be the space of generated edit records and u_0 the canonical identity edit. A valid record retains the cache or supplies a replacement box. For a valid edit, the deterministic transition $\Phi(x, u)$ changes only b, with $\Phi(x, u_0) = x$.

Definition 1 (Referent-preserving edit)**.** For an admissible task, a valid cache edit keeps d, s, z fixed and preserves the intended object and target answers under the same subsequent-query protocol:

$$o^\star(\Phi(x,u)) = o^\star(x), \qquad \mathbf{a}^\star(\Phi(x,u)) = \mathbf{a}^\star(x). \tag{2}$$

This invariance concerns the task definition; the revised box can still be incorrect. An external reselection defines a new task and requires relabeling (Appendix E). Invalid edits enter a deterministic failure state, as formalized in Appendix A and implemented in Appendix D.

3.2 Consistency and Recovery

Preserving a correct reference and recovering from an incorrect one are distinct learning goals. Let C_t indicate whether the active reference identifies the intended object at submitted checkpoint t. Define

$$p_{\text{ent}} = \mathbb{P}(C_{t+1} = 0 \mid C_t = 1), \qquad p_{\text{rec}} = \mathbb{P}(C_{t+1} = 1 \mid C_t = 0). \tag{3}$$

Error prevention lowers p_{ent}, whereas recovery raises p_{rec}; an improvement in one does not determine the other. We estimate the rates on shared starting-state strata, excluding intentional target changes. Appendix B.4 gives a two-state interpretation under additional assumptions.

Incentive conflict. Consider a binary keep/repair model with a fixed task-utility gain ΔU from repair. A citation term favors keeping by $d_c > 0$ with weight $\alpha \geq 0$, giving reward difference $\delta = \Delta U - \alpha d_c$.

Proposition 1 (Direction of a binary group-normalized update)**.** Sample $G \geq 2$ independent Bernoulli actions with repair probability $\omega = \sigma(\ell) \in (0,1)$ and rewards $R(u) = R_0 + \delta u$. With detached, group-centered advantages normalized by the population standard deviation plus $\epsilon_A > 0$, the unclipped score gradient at the rollout parameters, without KL, is

$$g(w) = \frac{w(1-w)\delta}{|\delta|\sqrt{w(1-w)} + \epsilon_A}, \tag{4}$$

where w is the sampled repair fraction. For $\delta \neq 0$, $\mathbb{E}[g(w)]$ has the sign of δ.

Thus $0 < \Delta U < \alpha d_c$ makes the update oppose a task-beneficial repair in this model. Appendix B gives the proof and its scope.

This condition motivates evaluating edits through downstream utility while preserving the task's referent. In a deployed citation scheme, the conflict depends on the correction path, update timing, and relative rewards; a model may cite an old region while correcting it. Our evaluation varies citation weight at shared starting states to test these dependencies. Literal citation rewards can favor agreement with an incorrect reference (Appendix B.2).

3.3 Learning from Downstream Utility

We train the model to revise references using the utility of subsequent answers. Each valid edit preserves the task definition established in Section 3.1, and its value depends on reasoning from the resulting state. Given state x, the policy generates an edit u followed by a continuation y:

$$P_\theta(u, y \mid x) := \pi^r_\theta(u \mid x)\, \pi^c_\theta(y \mid \Phi(x,u)). \tag{5}$$

The revision policy π^r_θ and continuation policy π^c_θ share one VLM in RP^2O. The executor receives the canonical edited state $\Phi(x,u)$ and follows the remaining query protocol. Its input excludes the editor's discarded reasoning, proposed answer, and any correction flag, so the edited reference is the observable intervention between the two stages. Each task has a fixed external query schedule ξ. Continuation laws and utilities below are conditional on this schedule, with ξ suppressed in the notation. The policy sees queries only when they are revealed; $x \sim \mu$ denotes sampling a start with its associated task record.

For a fixed horizon H, define the submitted-answer utility and joint objective:

$$\begin{aligned} U_x(u,y) &:= \frac{1}{H}\sum_{t=1}^{H} v(y_t, a^\star_t(x)) - \eta c(u,y), \\ J(\theta) &:= \mathbb{E}_{x\sim\mu}\mathbb{E}_{(u,y)\sim P_\theta(\cdot|x)}[U_x(u,y)]. \end{aligned} \tag{6}$$

The verifier $v \in [0, 1]$ checks submitted answers; missing turns score zero. Cost c counts declared resources. The start distribution μ is fixed within each local update. The mechanism study sets $\eta = 0$ and accounts for generated rollout tokens. Reference-IoU supervision provides a comparator for learning from downstream task utility. Evaluation additionally checks the maintained reference, distinguishing answer success from reference recovery.

3.4 Credit Assignment from the Joint Objective

We derive both credit signals from J, first using conditional expected utilities and then replacing them with rollout estimates. Freeze rollout parameters at $\bar{\theta}$ and define

$$\begin{aligned} Q^{\bar{\theta}}(x, u) &:= \mathbb{E}_{y \sim \pi^c_{\bar{\theta}}(\cdot|\Phi(x,u))}[U_x(u, y)], \\ \Delta^{\bar{\theta}}(x, u) &:= Q^{\bar{\theta}}(x, u) - Q^{\bar{\theta}}(x, u_0). \end{aligned} \tag{7}$$

The unchanged-state control executes the identity edit, $\Phi(x, u_0) = x$. Candidate and control branches share the task, query schedule, and executor parameters. For valid edits, continuation inputs differ only through the canonical edited reference; continuations use independent random draws.

Ideal credit decomposition. Let $S^r = \nabla_\theta \log \pi^r_\theta(u \mid x)|_{\theta=\bar{\theta}}$ and $S^c = \nabla_\theta \log \pi^c_\theta(y \mid \Phi(x, u))|_{\theta=\bar{\theta}}$. Differentiating the joint policy in Eq. (5) yields a score term for each stage. The conditional mean of each score is zero, so subtracting $Q^{\bar{\theta}}(x, u_0)$ from revision utility and $Q^{\bar{\theta}}(x, u)$ from continuation utility gives

$$\nabla J(\bar{\theta}) = \mathbb{E}\left[\underbrace{\Delta^{\bar{\theta}}(x, u) S^r}_{\text{edit decision}} + \underbrace{\big(U_x(u, y) - Q^{\bar{\theta}}(x, u)\big) S^c}_{\text{execution under the edit}}\right]. \tag{8}$$

The expectation samples starts from μ and actions from $P_{\bar{\theta}}$. The first term compares an edit with retaining the reference; the second compares a continuation with the mean utility under that edit. Both terms are required when the two policies share parameters. Appendix C gives the conditions and derivation.

Rollout estimates. Sample $G \geq 1$ edits, $K \geq 2$ independent continuations per edit, and $K_0 \geq 1$ independent control continuations. With utilities U_{ik} and U_{0k}, revision credit is

$$\widehat{Q}_i = \frac{1}{K} \sum_k U_{ik}, \quad \widehat{Q}_0 = \frac{1}{K_0} \sum_k U_{0k}, \quad A^r_i = \widehat{Q}_i - \widehat{Q}_0. \tag{9}$$

For execution, estimate the same-state baseline using the other continuations:

$$A^c_{ik} := U_{ik} - \frac{1}{K - 1} \sum_{k' \neq k} U_{ik'}. \tag{10}$$

The leave-one-out baseline is conditionally independent of the continuation it scores. All outcomes are retained, and all credit weights are detached. The forced identity edit supplies only a baseline; even a sampled candidate with $u_i = u_0$ uses independent control continuations.

Write S^r_i and S^c_{ik} for the scores evaluated on u_i and y_{ik}. Combining these estimates of the ideal credits gives

$$\widehat{g} = \frac{1}{G} \sum_i \left[A^r_i S^r_i + \frac{1}{K} \sum_k A^c_{ik} S^c_{ik}\right]. \tag{11}$$

Proposition 2 (Local two-stage score estimator)**.** Under the fixed-task, policy-support, sampling, and differentiation conditions in Assumptions 1–3, with $G \geq 1$, $K \geq 2$, and $K_0 \geq 1$, Eq. (11) satisfies $\mathbb{E}[\widehat{g}] = \nabla J(\bar{\theta})$. The two policies may share parameters.

The proof in Appendix C applies the score identity to the independent control and conditional leave-one-out baselines. This is a local gradient-estimation result, not a guarantee of monotonic improvement under practical updates.

Table 1: **Reference recovery and referent preservation.** Scores (%) are means with subscripted $\pm$ sample SD over three seeds. All methods use the shared cohort in Table 6; $L = H = 3$. Bold compares editing methods. R measures reference recovery; JR also requires the endpoint answer. The citation rows adapt RegionReasoner-style reward; w/o citation removes only that term. No edit reuses the $\mathrm{RP}^2\mathrm{O}$ executor.

Method	Answer ↑	R ↑	JR ↑	Pair ↑	Def. ↑	Damage ↓
No edit	$16.96_{\pm 0.68}$	-	-	-	-	-
w/ citation	$55.84_{\pm 3.01}$	$37.85_{\pm 0.44}$	$32.60_{\pm 1.07}$	$27.07_{\pm 1.18}$	$93.20_{\pm 1.27}$	$\mathbf{2.55}_{\pm 0.22}$
w/o citation	$56.11_{\pm 0.66}$	$48.94_{\pm 1.12}$	$41.07_{\pm 0.95}$	$32.01_{\pm 1.75}$	$85.81_{\pm 1.87}$	$8.20_{\pm 0.70}$
Re-grounding	$62.44_{\pm 0.32}$	$65.05_{\pm 0.42}$	$55.15_{\pm 0.60}$	$39.81_{\pm 2.54}$	$82.82_{\pm 2.28}$	$13.11_{\pm 1.68}$
SFT	$65.39_{\pm 0.81}$	$64.85_{\pm 1.25}$	$56.41_{\pm 0.59}$	$47.74_{\pm 0.63}$	$94.72_{\pm 0.62}$	$5.08_{\pm 0.34}$
GRPO (Answer)	$\mathbf{76.46}_{\pm 1.18}$	$71.23_{\pm 0.66}$	$66.05_{\pm 1.03}$	$56.73_{\pm 1.29}$	$93.81_{\pm 1.17}$	$5.33_{\pm 0.43}$
GRPO (IoU)	$68.95_{\pm 1.71}$	$75.80_{\pm 1.59}$	$63.72_{\pm 1.30}$	$52.99_{\pm 1.71}$	$\mathbf{96.56}_{\pm 0.57}$	$4.66_{\pm 0.34}$
$\mathbf{RP^2O}$ (Ours)	$75.68_{\pm 0.53}$	$\mathbf{78.23}_{\pm 0.72}$	$\mathbf{72.58}_{\pm 0.52}$	$\mathbf{65.68}_{\pm 1.10}$	$96.39_{\pm 0.32}$	$3.76_{\pm 0.60}$

Preserving the control comparison. We retain A_i^r without group re-centering: subtracting the candidate-group mean would cancel the shared control (Lemma 1). The detached, action-independent baseline leaves the expected gradient of J unchanged (Li et al., 2024). Its empirical contribution is assessed at common generated-token budgets.

3.5 Training and Inference

We implement the joint update with a PPO-style clipped surrogate (Schulman et al., 2017), using A_i^r on revision tokens and A_{ik}^c on continuation tokens. Likelihood ratios use the actual generation contexts. Token contributions are summed, continuations are averaged within each edit, and edits and starts are averaged equally. Appendix D specifies the surrogate, frozen-reference KL penalty, masks, and Algorithm 1.

At $\theta = \bar{\theta}$ and with the KL coefficient $\beta = 0$, the surrogate derivative equals $\widehat{g}$ (Corollary 1). Clipping away from that point and KL regularization lie outside this identity; the actual optimizer update is not itself asserted to be an unbiased gradient of J.

We train on correct histories, valid estimate corruptions, and coherent model-generated errors, sampling new revision decisions and continuations with history tokens used solely as context. Inference consists of one revision decision, which may retain the cache, followed by the scheduled continuation, without control rollouts or test labels.

4 Experiments

4.1 Experimental Setup

The primary backbone is Qwen2.5-VL-7B-Instruct (Bai et al., 2025). We follow the RegionReasoner protocol (Sun et al., 2026), with one revision decision followed by three scheduled questions ($H = L = 3$). Shared starts compare recovery from common errors; method-specific histories assess performance on each model's own trajectories. Interventions that supply retained, edited or annotated caches to frozen executors distinguish revision effects from execution differences. Same-backbone training comparisons share observable inputs and annotation access; native checkpoints provide task context. Core comparisons report means and sample SD over three seeds. Training budgets count all generated rollout tokens, including control branches. Further details on the comparisons are provided in Table 8.

4.2 Recovery and Referent Preservation

$\mathrm{RP}^2\mathrm{O}$ improves reference recovery and joint correctness while maintaining endpoint answer accuracy comparable to GRPO (Answer) (Table 1). On common corrupted starts, it achieves the highest R and JR among editing methods, with a 6.53-point JR gain over GRPO (Answer) and endpoint Answer

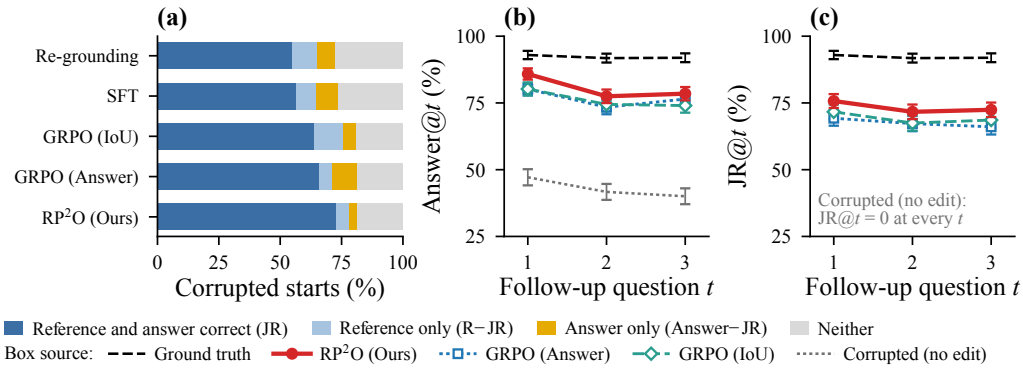


Figure 2: **Follow-up reasoning after one reference edit.** (a) Endpoint outcomes on the shared corrupted starts ($L = 3$; Table 1): both reference and answer correct (JR), reference correct only (R−JR), answer correct only (Answer−JR), and neither correct. (b, c) Answer@t and JR@t at each scheduled question under one frozen GRPO (Answer) executor, using annotated boxes (Ground truth), boxes from each editor, or retained corrupted boxes (no edit). Each cache is set once and held fixed throughout the continuation. Error bars are 95% image-cluster bootstrap intervals over the fixed cohort. For shared conditions, the averages over t and endpoint values match the GRPO (Answer) executor column of Table 9; the No-edit row of Table 1 uses the $\mathrm{RP}^2\mathrm{O}$ executor.

within one point. Decomposing the same outcomes (Figure 2a) shows that correct answers with incorrect references fall from 10.41% to 3.10%. The main gain is therefore that correct answers more often coincide with recovery of the maintained reference.

Recovery also requires preserving the information that defines the intended object. Semantic pairs test this requirement using the same image and candidate box under different task definitions. A shared candidate box can be an incorrect estimate in one task and a valid selection in the other. Pair accuracy therefore tests whether the model conditions reference maintenance on the information that defines the intended object. $\mathrm{RP}^2\mathrm{O}$ improves Pair by 8.95 points over GRPO (Answer) and attains 96.39% defining-selection compliance. Always re-grounding can recover erroneous references but damages 13.11% of correct starts, compared with 3.76% for $\mathrm{RP}^2\mathrm{O}$. These contrasts support selective revision that corrects inherited errors while respecting the task definition.

4.3 Reference Revision and Subsequent Reasoning

The revised caches contribute to the joint-recovery gain. Supplying each editor's outputs to both frozen executors separates the effect of the supplied state from differences in executor parameters (Table 9). With either executor held fixed, $\mathrm{RP}^2\mathrm{O}$ edits improve JR by 6.2–6.4 points and Avg. Answer by 3.9–5.3 points over GRPO (Answer) edits. These gains extend to the GRPO (Answer) executor, which attains higher Avg. Answer on both sets of learned caches. This supports a distinct contribution from the revised references.

One revision also benefits the full scheduled continuation. With the cache set once and then held fixed, $\mathrm{RP}^2\mathrm{O}$ edits yield the highest Answer@t and JR@t among edited caches at all three questions under the frozen GRPO (Answer) executor (Figure 2b, c). The JR advantage over GRPO (Answer) edits remains 4.41–6.45 points across positions. This persistence on a fixed cohort shows that the revised reference remains useful beyond the first follow-up. The annotated-cache condition remains ahead, leaving further scope to improve recovery.

4.4 Citation Incentives

Literal-citation rewards introduce a recovery–preservation tradeoff in the adapted baseline. Adding the RegionReasoner-style citation term reduces Damage from 8.20% to 2.55%, but lowers reference recovery by 11.09 points at similar endpoint Answer (Table 1). The weight sweep clarifies this tradeoff (Figure 4): increasing α from 0 to 2 reduces recovery across all three error sources and lowers the pooled `REBIND` rate on both corrupted starts (69.6% to 38.3%) and correct starts (13.2%

Table 2: **Supervision and credit assignment.** Scores (%) are means with sample SD over three seeds. Avg. Answer averages all three follow-ups; endpoint metrics of rows shared with Table 1 come from the same runs. A fixes the $\mathrm{RP}^2\mathrm{O}$ estimator and varies utility; B fixes task utility and varies credit. Revision control is $\widehat{Q}_0$; execution control is the within-state leave-one-out term. Bold marks within-panel optima.

Variant	Avg. Answer ↑	R ↑	JR ↑	Pair ↑	Damage ↓
A. Utility: ablations of $\mathbf{RP^2O}$					
$\mathbf{RP^2O}$ (Ours)	$\mathbf{75.81}_{\pm 0.61}$	$78.23_{\pm 0.72}$	$\mathbf{72.58}_{\pm 0.52}$	$\mathbf{65.68}_{\pm 1.10}$	$3.76_{\pm 0.60}$
w/ IoU utility	$72.90_{\pm 1.17}$	$\mathbf{79.04}_{\pm 1.53}$	$67.70_{\pm 1.22}$	$58.35_{\pm 1.93}$	$\mathbf{3.14}_{\pm 0.71}$
B. Credit assignment					
GRPO (Answer)	$76.68_{\pm 0.55}$	$71.23_{\pm 0.66}$	$66.05_{\pm 1.03}$	$56.73_{\pm 1.29}$	$5.33_{\pm 0.43}$
$\mathbf{RP^2O}$ (Ours)	$75.81_{\pm 0.61}$	$\mathbf{78.23}_{\pm 0.72}$	$\mathbf{72.58}_{\pm 0.52}$	$\mathbf{65.68}_{\pm 1.10}$	$3.76_{\pm 0.60}$
w/o revision control	$\mathbf{76.83}_{\pm 0.54}$	$76.56_{\pm 0.94}$	$71.40_{\pm 0.88}$	$64.87_{\pm 0.77}$	$\mathbf{3.66}_{\pm 0.85}$

to 4.1%). For this reward adaptation, lower damage accompanies suppressed correction, consistent with a preference for retention and the incentive conflict in the binary analysis.

4.5 Utility and Credit Assignment

Downstream answer utility favors revisions that support subsequent task success. Table 2A holds the interface, estimator, starts and sampling allocation fixed while replacing answer utility with mean IoU over the active cache and answer boxes. Using answer utility improves Avg. Answer by 2.91 points, JR by 4.88 points and Pair by 7.33 points; IoU utility yields slightly higher R and lower Damage. The distinction is informative: more accurate localization alone need not translate into greater joint correctness. Here, Avg. Answer measures correctness over the full continuation and directly matches the training utility at $\eta = 0$ (Eq. (6)).

The two-stage credit assignment further strengthens joint recovery under the same answer utility (Table 2B). Full $\mathrm{RP}^2\mathrm{O}$ achieves the highest R, JR and Pair, and the revision control adds 1.18 JR points over the variant that retains the editor but omits this control. The JR ordering persists when all candidate and control branches count toward a common generated-token budget (Figure 5). At the largest shared budget, 60.4M tokens, JR is 72.6% for $\mathrm{RP}^2\mathrm{O}$, 71.4% without revision control and 66.3% for GRPO (Answer). The advantage centers on joint recovery: GRPO (Answer) retains higher Avg. Answer at this budget (77.2% vs. 75.8%), underscoring the distinct information conveyed by answer utility and reference-aware evaluation.

4.6 Robustness and Grounding Performance

The recovery advantage extends beyond wrong-object cache corruptions (Table 7). $\mathrm{RP}^2\mathrm{O}$ improves JR over GRPO (Answer) by 8.49 points on coherent erroneous histories and 7.39 points on natural errors, and retains the highest JR and Pair under held-out wording. On each model's own trajectories, it starts with a lower fraction of correct references (75.06% vs. 78.65%) yet reaches higher JR@L (84.05% vs. 82.28%) at similar Avg. Answer. The final advantage therefore cannot be explained by a higher initial-reference accuracy. These trajectory results complement recovery comparisons on shared starts (Section 3.2).

We evaluate multi-turn grounding on RegionDial-Bench and single-turn localization on REC benchmarks (Table 3), preserving their native inputs, labels and scorers. Among same-backbone training methods, $\mathrm{RP}^2\mathrm{O}$ achieves the highest RegionDial AP on both splits, exceeding GRPO (Answer) by 1.93 and 1.17 points, while REC accuracy remains comparable to the other trained variants. These results show that improved reference maintenance is compatible with strong performance on the standard grounding tasks. Native checkpoints provide additional context, with $\mathrm{RP}^2\mathrm{O}$ ahead of RegionReasoner-7B on RegionDial AP and RegionReasoner-7B ahead on REC.

Table 3: **Multi-turn grounding and single-turn localization.** Native rows show means; training rows include subscripted ± sample SD over three seeds. Bold marks the best mean per column within each panel. Following the RegionReasoner protocol, RegionDial uses native detection AP and aggregation; REC uses Acc@0.5 (Appendix G.6).

	RegionDial AP ↑		REC Acc@0.5 ↑	
Method	RefCOCO+	RefCOCOg	RefCOCO+	RefCOCOg
Native checkpoints				
Qwen2.5-VL-7B	49.73	49.56	87.98	86.03
VisionReasoner-7B	74.29	73.36	87.65	87.30
RegionReasoner-7B	**80.70**	**78.51**	**88.53**	**88.58**
Same-backbone training				
SFT	$79.55_{\pm 0.89}$	$77.26_{\pm 0.87}$	$87.98_{\pm 0.32}$	$87.55_{\pm 0.30}$
GRPO (Answer)	$81.03_{\pm 0.36}$	$78.50_{\pm 0.19}$	$87.88_{\pm 0.09}$	$87.77_{\pm 0.23}$
RP^{2}O (Ours)	$\mathbf{82.96}_{\pm 0.21}$	$\mathbf{79.67}_{\pm 0.22}$	$\mathbf{88.32}_{\pm 0.17}$	$\mathbf{87.80}_{\pm 0.10}$

4.7 Inference Cost

The recovery gains require one additional inference call and about 20 generated tokens per dialogue. Total latency increases from 7.65 s without editing to 8.30 s for RP^{2}O, comparable to always re-grounding at 8.36 s (Table 10). Thus, selective revision improves preservation over unconditional re-localization at similar measured inference cost. These measurements cover one revision decision and all three follow-up answers; control rollouts are used only during training.

5 Discussion

The experiments distinguish consistency of the intended referent from persistence of a cached localization. Recovery and semantic-pair comparisons assess correction and referent preservation, while frozen-executor interventions show that the revised caches contribute to subsequent success. The citation-reward study illustrates why retaining a cache can reduce both damage and recovery. Together, these findings motivate evaluating consistency through the intended referent and the answers that depend on it. Reporting answer utility alongside joint recovery makes this contribution visible even when answer accuracy alone changes little.

These results show that downstream answer supervision can support reference recovery without a direct reward on cache accuracy. Revision credit measures improvement over retaining the cache, while execution credit compares continuations under the same edited state. Both signals therefore serve a common task objective. The frozen-executor comparisons further connect these recovery gains to the revised state supplied to subsequent reasoning.

The present evidence concerns an identifiable intended referent, one active reference and one revision opportunity before a fixed continuation. Extending the approach to repeated revision, adaptive dialogue, multiple references and other backbones remains an open direction.

6 Conclusion

We introduced RP^{2}O for learning to revise visual references while preserving the intended referent. By separating task-defining information from cached localization, it jointly trains revision and subsequent reasoning using downstream answer utility and distinct credit signals. Controlled evaluations show improved reference recovery, referent preservation and joint correctness, with endpoint answer accuracy comparable to answer-reward GRPO. Frozen-executor comparisons further support the contribution of the revised caches. These findings identify reference maintenance as a learnable component of multi-turn visual reasoning: consistency depends on preserving the intended object while allowing its estimated location to change.

AI USE STATEMENT

In this work, we used generative AI tools to assist in formulating mathematical claims and in drafting and revising proofs. We did not use generative AI tools to generate synthetic datasets, clean or reformat datasets, design experiments, or interpret experimental results. Additionally, we used generative AI tools to aid in refining the language and correcting grammatical errors in the manuscript. The authors reviewed all AI-assisted work. In particular, we checked the assumptions and validity of the mathematical claims, examined the proof steps for logical correctness, and checked the proofs for gaps or missing justifications. We revised the AI-assisted mathematical content where necessary. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. We collect no new human data and recruit no annotators. All images, expressions and object annotations come from public resources used under their original licences: the RefCOCO family and MSCOCO. We do not infer demographics, identities or other sensitive attributes, and references are indexed by sequential identifiers independent of any persistent identity. We release only annotations and dataset image identifiers, not images. The method is intended for studying reference maintenance in visual dialogue, not for surveillance or biometric identification, and we discourage applying it to private imagery without consent. We are not aware of any conflicts of interest.

REPRODUCIBILITY STATEMENT

All experiments use publicly available datasets and the publicly available Qwen2.5-VL-7B-Instruct backbone. Appendix D and Table 5 give the full training and inference configuration, including the optimizer, the rollout and decoding settings, the hardware, and the three random seeds shared by every run. Appendix E describes task construction, the image-level split, and the aggregation conventions behind every reported mean and standard deviation; Appendix F gives the complete prompts and a worked example; and Appendices A–C state the assumptions and give complete proofs.

REFERENCES

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*, 2025. URL https://arxiv.org/abs/2502.13923.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-Adv: Exploring MLLM visual reasoning in referring expression tasks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ac37a79cc303ca982703f14caa033d8b-Abstract-Conference.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1544. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Zhaolin Gao, Wenhao Zhan, Jonathan D. Chang, Gokul Swamy, Kianté Brantley, Jason D. Lee, and Wen Sun. Regressing the relative future: Efficient policy optimization for multi-turn RLHF. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/186094ef879dcbd5ef6879782916c309-Abstract-Conference.html.

Haojie Huang, Xinlei Yu, Chengming Xu, Zhangquan Chen, Cheng Yang, Qingdong He, Yu Yang, Jiangning Zhang, and Xiaobin Hu. Evidence-RL: Towards evidence-intensive visual reasoning. *arXiv preprint arXiv:2608.08021*, 2026a. URL https://arxiv.org/abs/2608.08021.

Xinyu Huang, Yuhao Dong, Weiwei Tian, Bo Li, Rui Feng, and Ziwei Liu. MGPO: Thinking with images via multi-turn grounding-based reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 383–399. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-acl.18. URL https://aclanthology.org/2026.findings-acl.18/.

Muhammed Emrullah Ildiz, Halil Alperen Gozeten, Ege Onur Taga, and Samet Oymak. Learning to correct: Reinforcement learning for multi-attempt chain-of-thought. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2hGL0nGPLu.

Yue Jiang, Xue Jiang, Lihua Zhang, Zhiqiang Wang, Yuhang Lu, Peng Wang, Bo Han, Feng Zheng, and Dingkang Yang. MM-Snowball: Evaluating and mitigating hallucination snowballing in multimodal multi-turn dialogue. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vL1xAHOUmd.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/871ac99fdc5282d0301934d23945ebaa-Abstract-Conference.html.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/59f6421e64707225fdf5b28840679a07-Abstract-Conference.html.

Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. Visual self-refine: A pixel-guided paradigm for accurate chart parsing. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/89b0e466b46292ce0bfe53618aadd3de-Abstract-Conference.html.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29128–29163. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24cd.html.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

Geng Liu, Fei Zhu, Rong Feng, Changyi Ma, Shiqi Wang, and Gaofeng Meng. Intent mismatch causes LLMs to get lost in multi-turn conversation. *arXiv preprint arXiv:2602.07338*, 2026a. URL https://arxiv.org/abs/2602.07338.

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*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/97c2f0fac182353062d304d0322ae285-Abstract-Conference.html.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–20, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Mao_Generation_and_Comprehension_CVPR_2016_paper.html.

Lei Peng, Shuai Lv, and Wei Hu. ReGround: Restoring visual grounding in multi-step reasoning through self-diagnosis and visual re-examination. *arXiv preprint arXiv:2608.04385*, 2026. URL https://arxiv.org/abs/2608.04385. Accepted to ACM Multimedia 2026.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J. Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5049. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ddbd83ac1ad27304a72b873124c2dac2-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Wenfang Sun, Hao Chen, Yingjun Du, Yefeng Zheng, and Cees G. M. Snoek. RegionReasoner: Region-grounded multi-round visual reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8296d5800a8e68e58ad0472b393be80e-Abstract-Conference.html.

Liyan Tang, Fangcong Yin, and Greg Durrett. Visually grounded self-reflection for vision-language models via reinforcement learning. *arXiv preprint arXiv:2607.02490*, 2026. URL https://arxiv.org/abs/2607.02490. Version 1.

Animesh Tripathy and Aswanth Krishnan. Iterative visual thinking and the self-correction mirage in VLM grounding. *arXiv preprint arXiv:2606.13156*, 2026. URL https://arxiv.org/abs/2606.13156. Version 2, revised July 6, 2026.

XuDong Wang, Shaolun Zhang, Shufan Li, Konstantinos Kallidromitis, Kehan Li, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. SegLLM: Multi-round reasoning segmentation with large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/8e125fb53198f2143716d969584d0a28-Abstract-Conference.html.

Zhengxiang Wang and Owen Rambow. When seeing is not enough: Benchmarking interactive visual grounding in LVLMs. *arXiv preprint arXiv:2608.23978*, 2026. URL https://arxiv.org/abs/2608.23978. Accepted to EMNLP 2026.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. Perception-aware policy optimization for multimodal reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/82846e19e6d42ebfd4ace4361def29ae-Abstract-Conference.html.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, ChengXiang Zhai, and Klara Nahrstedt. VTool-R1: VLMs learn to think with images via reinforcement learning on multimodal tool use. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7e6f445a74cdb71931aac64f1e3f49c9-Abstract-Conference.html.

Tianrun Xu, Haoda Jing, Ye Li, Yuquan Wei, Jun Feng, Guanyu Chen, Haichuan Gao, Tianren Zhang, Jing Liu, and Feng Chen. DeFacto: Counterfactual thinking with images for enforcing evidence-grounded and faithful reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=TcICHju9Mo.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. TreeRPO: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*, 2025. URL https://arxiv.org/abs/2506.05183. Version 2.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *Computer Vision – ECCV 2016*, volume 9906 of *Lecture Notes in Computer Science*, pp. 69–85. Springer, 2016. doi: 10.1007/978-3-319-46475-6_5. URL https://link.springer.com/chapter/10.1007/978-3-319-46475-6_5.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-GRPO: Scaffolded group relative policy optimization for enhancing LLM reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d604a371bf833227ef4f0daf3c1c8135-Abstract-Conference.html.

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. Rank-GRPO: Training LLM-based conversational recommender systems with reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fb8f2d4e90018c0fae060dff09a91ce3-Abstract-Conference.html.

A Reference Semantics and Identifiability

Definition 2 (Admissible task instance)**.** A task instance is admissible if the available description or selection identifies a unique target under the declared task, the scheduled reference-dependent queries have validated answers, and a cache-only intervention preserves the task and the information needed to recover its intended target. The semantic role is observable, either explicitly in the controlled protocol or through ordinary dialogue. The external query schedule is fixed before candidate sampling, independently of its success.

Task invariants and failure states. Definition 1 applies to valid edits of an admissible task. It expresses invariance of the intended object and answer labels, not correctness of the edited localization. With $\mathcal{X}$ denoting ordinary observable states, define the total transition

$$\Phi : \mathcal{X} \times \mathcal{U} \longrightarrow \mathcal{X} \cup \{\perp\}. \tag{12}$$

For a valid edit, Φ updates the cache and its designated active aliases while preserving the task-defining fields. An invalid record maps to $\perp$, a terminal failure state whose continuation is a fixed output $y_\perp$: $\pi^c_\theta(y_\perp \mid \perp) = 1$ and its score is zero. All answer scores are then zero, and $U_x(u, y_\perp) = -\eta c(u, y_\perp)$ retains the generated edit cost. The invalid edit remains a sampled revision action. Appendix D gives the parser and resource conventions.

Table 4: **Notation.** State, policy and sampling symbols used in the derivations.

Symbol	Meaning
State and reference	
$x = (I, h, r)$	Observable image, history and reference state.
$r = (d, s, b, z)$	Originating description, optional selection, cached box, semantic role.
E, D	Estimate and definition roles.
$\mathcal{U}$, u_i	Space of generated edit records and sampled edit i.
u_0	Canonical identity edit, with $\Phi(x, u_0) = x$.
$\Phi(x, u)$	Canonical edited state, or $\perp$ for an invalid edit; common transition.
Policies and objective	
$\bar{\theta}, \theta$	Frozen rollout parameters and current optimization parameters.
$\pi^r_\theta, \pi^c_\theta$	Revision and continuation policies.
π^a_{ref}	Frozen KL reference policy for stage $a \in \{r, c\}$.
$Q, \widehat{Q}$	Conditional expected utility and Monte Carlo estimate.
Δ, A^r_i, A^c_{ik}	Expected utility gain, revision credit, and execution credit.
$\mathbf{a}^\star$	Validated target answers, used by training/evaluation only.
μ, ξ	Fixed local distribution of start/task records and external query schedule.
Sampling and evaluation	
y_{ik}	Fresh continuation k under candidate edit i.
t, ℓ	Dialogue-turn and generated-token indices.
H, L	Fixed continuation horizon and submitted evaluation checkpoint.
G, K, K_0	Counts of edits, continuations per edit, and control continuations.

Table 4 distinguishes the submitted checkpoint L from sampling count K, and the rollout snapshot from the independently frozen KL reference policy. Semantic roles identify the source of the referent, not whether a cached box is correct.

Proposition 3 (Indistinguishable roles)**.** Suppose two equally likely admissible tasks have identical policy-visible observations after role and intent cues are removed, but require different deterministic answers $a_0 \neq a_1$. Then no policy using only those observations has mean exact-answer accuracy greater than $1/2$ on the pair.

Proof. Let O remove the specified role and intent cues, so that $O(x_0) = O(x_1) = o$. The policy has a common answer distribution $p(a \mid o)$ on the pair. Mean success is $\frac{1}{2}[p(a_0 \mid o) + p(a_1 \mid o)] \leq \frac{1}{2}$, since a_0 and a_1 are distinct outcomes of one distribution. The statement also covers random policies. □

The proposition makes observable task information a prerequisite for interpreting recovery behavior. Dialogue that disambiguates the role falls outside its indistinguishability premise. Held-out wording

tests whether role interpretation generalizes beyond explicit tags; the same user query is used for correct and incorrect estimates to avoid revealing cache correctness through the query.

Multiple references. The main derivation edits one active reference. The state interface stores a map of reference identifiers and immutable origins. The main study edits one protocol-selected reference per opportunity, followed by a continuation over the scheduled queries with no further explicit revision decisions in the measured horizon. A sequential extension can edit references in a declared order, with later decisions included in the continuation distribution and its likelihood. An autoregressive model can attend to the entire history, so missing edges in a symbolic dependency graph do not establish independent reference-wise returns. The graph supports evaluation controls; attention isolation requires explicit context restrictions.

Selection geometry. A defining reference denotes the selected object. Boundary refinement is admissible when it retains that identity, whereas a selection spanning multiple plausible objects may be ambiguous. The data protocol specifies these cases consistently for all methods and evaluates localization quality separately from preservation of the task definition.

B Consistency Incentives and Recovery

B.1 Proof of Proposition 1

The binary model isolates the direction induced by fixed rewards under group normalization. The following calculation also covers sampled groups containing only one action.

Proof. For a sampled group with m repairs, put $w = m/G$. Then

$$\overline{R} = R_0 + w\delta, \qquad \sigma_R = |\delta|\sqrt{w(1-w)}, \tag{13}$$

$$A(1) = \frac{(1-w)\delta}{\sigma_R + \epsilon_A}, \qquad A(0) = \frac{-w\delta}{\sigma_R + \epsilon_A}. \tag{14}$$

The Bernoulli score is $\partial_\ell \log \pi(u) = u - \omega$. Consequently,

$$\frac{1}{G}\sum_i A_i(u_i - \omega) = wA(1)(1-\omega) + (1-w)A(0)(-\omega) \tag{15}$$

$$= \frac{w(1-w)\delta}{|\delta|\sqrt{w(1-w)} + \epsilon_A}. \tag{16}$$

For $m = 0$ or G, this expression is zero. Every mixed group has the sign of δ, and has positive probability because $\omega \in (0,1)$ and $G \geq 2$. The expectation is therefore strictly of that sign. □

Scope of the binary model. The proposition characterizes reward-induced preference reversal for binary actions with fixed rewards at the rollout parameters. Extending it to a VLM requires accounting for stochastic outcomes, multiple revision proposals, active clipping, KL regularization, shared representations, and changes in the start distribution. The shared-start study in Section 4 tests how this mechanism relates to observed error entry and recovery.

B.2 Citation Incentives for a Single Reference

Let the required reference set be $B = \{b\}$. In the ICLR 2026 version of RegionReasoner (Sun et al., 2026, Eq. (4)), with its bbox keyword flag equal to one, the citation component is

$$C(S;B) = \left(1 + \frac{|S \cap B|}{\max(|S|,1)}\right)\begin{cases}1, & S \setminus B = \varnothing,\\ 1/2, & S \setminus B \neq \varnothing.\end{cases} \tag{17}$$

For a distinct $b' \notin B$:

$$C(\{b\};B) = 2, \qquad C(\{b'\};B) = 1/2, \qquad C(\{b,b'\};B) = 3/4. \tag{18}$$

Interpreting b' as a correct repair requires an estimate-role task, a valid original target, and a checked replacement box. A successful correction may also retain a citation to the old region. The calculation identifies a potential incentive conflict; its occurrence in a deployed system depends on reward timing and the correction paths actually used.

B.3 COMMON-BASELINE CANCELLATION

Lemma 1 (Shift invariance of group normalization)**.** For a fixed reward vector $(R_1, \ldots, R_G)$ and any common scalar b_0, using the same standard-deviation convention and stabilizer $\epsilon_A > 0$ gives

$$\frac{(R_i - b_0) - \mathrm{mean}_j(R_j - b_0)}{\mathrm{std}_j(R_j - b_0) + \epsilon_A} = \frac{R_i - \mathrm{mean}_j\, R_j}{\mathrm{std}_j\, R_j + \epsilon_A}. \tag{19}$$

Proof. $\mathrm{mean}_i(R_i - b_0) = \overline{R} - b_0$, and each centered residual is $(R_i - b_0) - (\overline{R} - b_0) = R_i - \overline{R}$. Their standard deviation is unchanged. Substitution yields the identity, including zero-variance groups when $\epsilon_A > 0$. □

The lemma remains true when b_0 is estimated from extra rollouts: condition on that realized estimate and it is still a common scalar. It need not hold for different per-candidate baselines, nonlinear reward transformations, or selective acceptance of trajectories; those are different algorithms and require separate analysis. The associated negative-control implementation should yield identical normalized advantages up to numerical precision.

B.4 ERROR ENTRY AND RECOVERY DYNAMICS

For a time-homogeneous two-state chain, assume the transition rates in Eq. (3) are constant and target identity does not change. If $e_t = \mathbb{P}(C_t = 0)$, the law of total probability gives

$$e_{t+1} = p_{\mathrm{ent}}(1 - e_t) + (1 - p_{\mathrm{rec}})e_t = p_{\mathrm{ent}} + (1 - p_{\mathrm{ent}} - p_{\mathrm{rec}})e_t. \tag{20}$$

When $p_{\mathrm{ent}} + p_{\mathrm{rec}} > 0$, the stationary error fraction is $e_{\mathrm{stat}} = p_{\mathrm{ent}}/(p_{\mathrm{ent}} + p_{\mathrm{rec}})$. Convergence to this value from every initial distribution holds when $0 < p_{\mathrm{ent}} + p_{\mathrm{rec}} < 2$. With no transitions, every initial distribution is stationary. With both rates one, nonstationary initial distributions alternate, while the stationary distribution remains fixed. These identities describe the stipulated two-state model. Empirical reporting uses matched strata; extrapolation to long-run dialogue error requires validating the constant-rate assumptions.

C POLICY-GRADIENT ANALYSIS

C.1 ASSUMPTIONS AND A BASELINE IDENTITY

We first condition on a task record (x, ξ), where x is the observable start and ξ fixes the external subsequent-query schedule, and then average task records under μ. As in the main text, ξ is suppressed in continuation-law and utility notation; the policy observes future queries only when they are revealed.

Assumption 1 (Task and policy)**.** For each local update, the task-record distribution μ, finite horizon, and generation cap are fixed. The policies are normalized and differentiable, and positive on fixed supports. The transition Φ is deterministic, and utility is bounded with no direct parameter dependence. A valid edit changes only the cached box. An invalid edit enters the failure state $\perp$, whose continuation is deterministic and has zero policy score; its answer rewards are zero and its generated edit cost is retained.

Assumption 2 (Rollout sampling)**.** All candidate edits and continuations are freshly sampled at fixed parameters $\bar{\theta}$. Conditional on (x, ξ), the $G \geq 1$ candidate edits are independent and identically distributed with law $\pi^r_{\bar{\theta}}(\cdot \mid x)$. Conditional also on all candidate edits, the $K \geq 2$ continuations per edit are mutually independent within and across branches, with law $\pi^c_{\bar{\theta}}(\cdot \mid \Phi(x, u_i))$ in branch i. The $K_0 \geq 1$ control continuations are independent and identically distributed with law $\pi^c_{\bar{\theta}}(\cdot \mid \Phi(x, u_0))$, independently of all candidate edits and continuations. Sharing policy parameters does not replace these independent random draws.

Assumption 3 (Differentiation conventions)**.** Rewards, estimated baselines, and sampled states are detached during optimization. Differentiation may be exchanged with the expectations below; finite support suffices, and more general support requires the corresponding regularity conditions.

Lemma 2 (Conditional score and baseline identity)**.** Let $a \sim p_\theta(\cdot \mid v)$ satisfy Assumptions 1 and 3, and write $S(a \mid v) = \nabla_\theta \log p_\theta(a \mid v)$. Then $\mathbb{E}[S \mid v] = 0$. If an integrable detached scalar B is conditionally independent of a given v, then

$$\mathbb{E}[BS \mid v] = \mathbb{E}[B \mid v] \sum_a \nabla_\theta p_\theta(a \mid v) = 0. \tag{21}$$

Proof. Normalization gives $\sum_a \nabla_\theta p_\theta(a \mid v) = \nabla_\theta 1 = 0$. Conditional independence factors $\mathbb{E}[BS \mid v]$ into $\mathbb{E}[B \mid v]\mathbb{E}[S \mid v]$. Taking $B = 1$ gives the first statement. □

C.2 Proof of Proposition 2

Proof. Fix (x, ξ) and suppress this conditioning where unambiguous. Write $\Phi_u := \Phi(x, u)$, $J_{x,\xi}(\theta) := \mathbb{E}[U_x(u, y) \mid x, \xi]$, and $Q^\theta(x, u) := \mathbb{E}_{\pi^c_\theta(\cdot \mid \Phi_u)}[U_x(u, y)]$.

Step 1: decompose the joint score. Differentiating the joint distribution gives

$$\begin{aligned}
\nabla J_{x,\xi}(\theta) &= \sum_{u,y} \nabla \left[\pi^r_\theta(u \mid x)\pi^c_\theta(y \mid \Phi_u)\right] U_x(u, y) \\
&= \mathbb{E}_{u,y}\left[U_x(u, y)\left(S^r(u) + S^c(y \mid u)\right)\right] \\
&= \mathbb{E}_u[Q^\theta(x, u)S^r(u)] + \mathbb{E}_{u,y}[U_x(u, y)S^c(y \mid u)].
\end{aligned} \tag{22}$$

Here both scores are evaluated at θ. By Lemma 2, subtracting $Q^\theta(x, u_0)$ in the revision term and $Q^\theta(x, u)$ in the continuation term changes neither expectation. At $\theta = \bar{\theta}$, averaging over task records gives the ideal centered decomposition in Eq. (8). The product rule remains valid when both policies share parameters.

Step 2: estimate revision credit. At $\bar{\theta}$, Assumption 2 gives $\mathbb{E}[\widehat{Q}_i \mid x, \xi, u_i] = Q^{\bar{\theta}}(x, u_i)$ and $\mathbb{E}[\widehat{Q}_0 \mid x, \xi, u_i] = Q^{\bar{\theta}}(x, u_0)$. Consequently,

$$\begin{aligned}
\mathbb{E}[A^r_i S^r_i \mid x, \xi] &= \mathbb{E}_u\Big[\big(Q^{\bar{\theta}}(x, u) - Q^{\bar{\theta}}(x, u_0)\big)S^r(u)\Big] \\
&= \mathbb{E}_u[Q^{\bar{\theta}}(x, u)S^r(u)],
\end{aligned} \tag{23}$$

where the last equality uses Lemma 2. This also covers a sampled candidate $u_i = u_0$, because its continuations and the forced control use independent draws.

Step 3: estimate execution credit. The leave-one-out baseline $B^c_{ik} := (K-1)^{-1} \sum_{k' \neq k} U_{ik'}$ is independent of y_{ik} conditional on (x, ξ, u_i). Thus

$$\mathbb{E}[B^c_{ik} S^c_{ik} \mid x, \xi, u_i] = \mathbb{E}[B^c_{ik} \mid x, \xi, u_i]\mathbb{E}[S^c_{ik} \mid x, \xi, u_i] = 0. \tag{24}$$

Averaging the remaining $U_{ik}S^c_{ik}$ terms over k and then u_i recovers the second term of Eq. (22). Averaging both terms over candidates and task records yields $\mathbb{E}[\hat{g}] = \nabla J(\bar{\theta})$. Both score terms contribute to this joint objective, including when the policies share parameters. □

Role of the control baseline. The unchanged-state control supplies a reference level for revision proposals while preserving the expected policy gradient through Lemma 2. Its role is to shape the finite-sample estimator; the objective remains J. Stopping gradients through the control is essential: differentiating both terms in $\mathbb{E}[Q^\theta(x, u) - Q^\theta(x, u_0)]$ would introduce the executor derivative of $Q^\theta(x, u_0)$ and change the objective. ReMax provides a relevant antecedent for response-based control variates (Li et al., 2024).

Leave-one-out execution credit. For independent identically distributed continuations conditional on (x, ξ, u), a self-including mean gives

$$\mathbb{E}\left[\frac{1}{K}\sum_k (U_k - \overline{U})S_k \,\middle|\, x, \xi, u\right] = \left(1 - \frac{1}{K}\right)\mathbb{E}[US \mid x, \xi, u]. \tag{25}$$

The leave-one-out form corrects this finite-sample factor. Sample-standard-deviation scaling introduces a nonlinear transformation outside Proposition 2. Success-only replay, reward-dependent rejection, and mismatched prompt likelihoods also alter its sampling or score assumptions.

C.3 Conditional Monte Carlo variance

Proposition 4 (Noise of a matched estimate)**.** Fix (x, ξ, u) and let S^r be the sampled edit's fixed score vector. Conditional on these quantities, suppose the $K \geq 1$ candidate continuation utilities are independent and identically distributed with finite variance σ_u^2, the $K_0 \geq 1$ control utilities are independent and identically distributed with finite variance σ_0^2, and the two groups are mutually independent. Then

$$\mathrm{Cov}\Big[(\widehat{Q}_u - \widehat{Q}_0)S^r \mid x, \xi, u\Big] = S^r(S^r)^\top \left(\frac{\sigma_u^2}{K} + \frac{\sigma_0^2}{K_0}\right). \tag{26}$$

Proof. Cross-group independence gives $\mathrm{Cov}(\widehat{Q}_u, \widehat{Q}_0 \mid x, \xi, u) = 0$. Within-group independence gives $\mathrm{Var}(\widehat{Q}_u \mid x, \xi, u) = \sigma_u^2/K$ and $\mathrm{Var}(\widehat{Q}_0 \mid x, \xi, u) = \sigma_0^2/K_0$. Expanding the variance of their difference and multiplying by $S^r(S^r)^\top$ proves the identity. Correlated branch draws would require retaining the cross-covariance term. □

This identity quantifies conditional sampling noise for one candidate. Total policy-gradient variance also depends on candidate variation, the cross-candidate covariance induced by a shared control, and covariance between the revision and execution terms. Estimating the control adds noise relative to a known baseline and may increase or decrease variance relative to using no baseline. At fixed cost, greater continuation precision also reduces candidate breadth. These dependencies motivate accounting for all sampling branches in the generated-token comparison in Appendix H.4.

Reward-to-go and likelihood conventions. A reward determined before an action acts as an action-independent baseline conditional on the preceding history. For the local gradient identity, normalized token conditionals define the sequence probability, and sequence scores retain sums over generated tokens. A length-normalized score changes trajectory weighting. These conventions determine which empirical variants are covered by Proposition 2.

D Optimization and Implementation Details

D.1 Token likelihoods and clipping

For generated revision token $u_{i,\ell}$, let $c^r_{i,\ell} = (x, u_{i,<\ell})$ be its actual context. For a generated continuation token $y_{ik,\ell}$, let $c^c_{ik,\ell}$ include $\Phi(x, u_i)$, preceding generated tokens, and only those environment observations available at that time. The two likelihood ratios are

$$\rho^r_{i,\ell}(\theta) = \frac{\pi^r_\theta(u_{i,\ell} \mid c^r_{i,\ell})}{\pi^r_{\tilde{\theta}}(u_{i,\ell} \mid c^r_{i,\ell})}, \qquad \rho^c_{ik,\ell}(\theta) = \frac{\pi^c_\theta(y_{ik,\ell} \mid c^c_{ik,\ell})}{\pi^c_{\tilde{\theta}}(y_{ik,\ell} \mid c^c_{ik,\ell})}. \tag{27}$$

Both probabilities evaluate the sampled action under identical grammar constraints. For clipping width $\epsilon > 0$ and stop-gradient operator sg, define

$$\psi_\epsilon(\rho, A) = \min\{\rho\,\mathrm{sg}(A), \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)\,\mathrm{sg}(A)\}. \tag{28}$$

With policy-generated token sets $\mathcal{T}^r_i$ and $\mathcal{T}^c_{ik}$, the per-start objective to maximize is

$$\begin{aligned}\widehat{\mathcal{J}}_{\mathrm{clip}}(\theta) = &\frac{1}{G}\sum_i \left[\sum_{\ell \in \mathcal{T}^r_i} \psi_\epsilon(\rho^r_{i,\ell}, A^r_i) + \frac{1}{K}\sum_k \sum_{\ell \in \mathcal{T}^c_{ik}} \psi_\epsilon(\rho^c_{ik,\ell}, A^c_{ik})\right] \\ &- \beta \widehat{\mathcal{D}}_{\mathrm{KL}}(\theta),\end{aligned} \tag{29}$$

where $\beta \geq 0$. The loss-minimization convention is $\mathcal{L} = -\widehat{\mathcal{J}}_{\mathrm{clip}}$. Policy-credit terms retain token sums, and starts are averaged equally; the KL reduction is specified next.

KL direction and reduction. For stage $a \in \{r, c\}$ and a sampled context c, define the exact categorical next-token divergence

$$\kappa_\theta^a(c) := \sum_{v \in \mathcal{V}^a(c)} \pi_\theta^a(v \mid c) \log \frac{\pi_\theta^a(v \mid c)}{\pi_{\text{ref}}^a(v \mid c)}, \tag{30}$$

where $\mathcal{V}^a(c)$ is the fixed allowed token support at that context. Both distributions are normalized on this support, and the independently frozen KL reference policy is positive wherever the current policy is positive. The divergence is from the current policy to the KL reference policy; gradients pass through the current distribution only.

For each start, the candidate and continuation weights follow Eq. (29), with weighted token count N_x:

$$\begin{aligned} N_x &:= \frac{1}{G} \sum_i \left[|\mathcal{T}_i^r| + \frac{1}{K} \sum_k |\mathcal{T}_{ik}^c| \right], \\ \widehat{\mathcal{D}}_{\text{KL}}(\theta) &:= \frac{1}{G N_x} \sum_i \left[\sum_{\ell \in \mathcal{T}_i^r} \kappa_\theta^r(c_{i,\ell}^r) + \frac{1}{K} \sum_k \sum_{\ell \in \mathcal{T}_{ik}^c} \kappa_\theta^c(c_{ik,\ell}^c) \right]. \end{aligned} \tag{31}$$

Every sampled edit contains at least one generated token, so $N_x > 0$. Contexts, token sets, and N_x are held fixed during the update, and per-start objectives are averaged equally over the batch. Input tokens, forced-control trajectories, and deterministic failure outputs are excluded from both KL sums and counts; generated tokens of an invalid edit remain included. The weighted token mean specifies the KL penalty only, while policy-credit terms retain their token sums. Per-trajectory normalization, a sampled KL approximation, or a different reduction defines a separately declared variant.

Corollary 1 (Local derivative of the surrogate)**.** For a fixed start and its sampled rollouts with detached credit weights, the token ratios in Eq. (27), token sums and stage weights in Eq. (29), $\epsilon > 0$, and $\beta = 0$,

$$\nabla_\theta \widehat{\mathcal{J}}_{\text{clip}}(\theta)\Big|_{\theta=\bar{\theta}} = \widehat{g}. \tag{32}$$

Under the assumptions of Proposition 2, the expectation of this derivative is $\nabla J(\bar{\theta})$.

Proof. At $\theta = \bar{\theta}$, every sampled-token ratio is one and clipping is inactive in a neighborhood. Its derivative is the sampled-token score. Summing over generated tokens gives the sequence score for each stage, while the factors $1/G$ and $1/K$ recover Eq. (11). Deterministic failure outputs have zero score and no generated-token contribution. Applying the same averaging over starts preserves the equality for a batch. □

Dividing each action's token sum by its own length changes relative weighting and is outside this equality. A common fixed batch scale changes only the scale of the gradient. The specified optimizer also uses a gradient-norm clip and KL regularization; neither is covered by the unbiasedness statement for J.

Sampling distribution. Both likelihoods use the context that actually generated each token. Prompt-aware ratios are also explicit in Scaf-GRPO (Zhang et al., 2026). Here context matching is paired with retaining all sampled edits and continuations, including failures. The forced control supplies a detached baseline; its generated tokens, stored observations, and gold annotations receive no direct policy loss. The formal experiment uses normalized full-support rollout decoding (temperature one, no top-k/top-p truncation). An alternative sampling policy defines a separate variant and enters its logged likelihood denominator.

D.2 Data, observations, and budget

A training start is a pre-edit state containing the image, permitted history, task-defining information and active cache, together with its scheduled three-question continuation. One data pass refers to the training-start collection; the G candidate edits and their sampled continuations are rollouts from a start, not additional data examples. The 1358 evaluation families in Table 6 describe the diagnostic cohort.

Algorithm 1 One local update of $\mathrm{RP}^2\mathrm{O}$

Require: Start batch from fixed μ; $G \geq 1$, $K \geq 2$, $K_0 \geq 1$; current policy θ
Freeze rollout parameters $\bar{\theta} \leftarrow \theta$
for each start x **do**
  Load an offline-audited task; provide no correctness flag to the policy
  Sample G edits $u_i \sim \pi^r_{\bar{\theta}}(\cdot \mid x)$
  Sample K_0 independent continuations from $\Phi(x, u_0)$; compute $\widehat{Q}_0$
  for each sampled u_i **do**
    Apply syntax-checked transition $x_i \leftarrow \Phi(x, u_i)$
    if $x_i = \perp$ **then**
      Record K deterministic failures, zero answer scores, and edit cost
    else
      Sample K fresh continuations from x_i using independent random streams
    end if
    Evaluate submitted answers and declared costs; retain all outcomes
    Compute A^r_i and A^c_{ik} by Eqs. (9) and (10)
  end for
end for
Detach advantage weights, observations, and rollout log-probabilities
Update θ with Eq. (29); mask all input tokens and all forced-control trajectories
Retain uncentered A^r; use forced u_0 rollouts only as a detached baseline

The start distribution combines clean references, valid estimate corruptions, and coherent mistakes sampled from earlier policy snapshots. Reference-role cues appear in every condition, including correct estimates, so they do not identify errors. Every baseline receives the same original visual and linguistic evidence from a replayed start, with fresh continuations sampled from that state. Gold boxes support task auditing and supervised comparison methods; they are excluded from deployment and branch selection.

The reported $\mathrm{RP}^2\mathrm{O}$ runs use $G = 4$, $K = 2$ and $K_0 = 2$, giving $GK + K_0 = 10$ continuation samples per start, plus four edit generations. Generated-token accounting includes every candidate and control continuation. We use the fixed settings in Table 5, without validation-based selection.

Invalid edits. A parser checks schema, finite coordinates, bounds, and reference identifiers. Semantic correctness is evaluated separately. The specification uses the failure state $\perp$ defined in Appendix A: all H answer utilities are zero, generated edit cost is retained, and the subsequent output is a fixed failure token with zero policy score. Such candidates remain in the sample without replacement. This rule operates without inspecting gold boxes.

Execution and exploration. Effective recovery requires sampling useful edits and an executor that uses the resulting state. Joint reference–answer evaluation measures whether a correct answer accompanies reference recovery, and component comparisons isolate editor and executor behavior. These behavioral tests do not establish internal reasoning faithfulness.

D.3 Connecting the ideal and implemented objectives

Proposition 2 establishes the score estimator's expectation at the rollout parameters. Corollary 1 connects it to the surrogate derivative with unit revision/continuation weights, no KL term, and sums over generated tokens. Both score terms contribute when the stages share parameters. The specified reduction averages continuation losses within an edit and edits within a start, preserving the relative stage weights in Eq. (29). Repeating the revision loss K times without compensating normalization would change those weights.

The KL anchor is an independently frozen reference checkpoint, distinct from the rollout snapshot $\bar{\theta}$, with the reduction in Eq. (31). A common fixed batch scale may be absorbed into the learning rate; per-trajectory length normalization and sample-standard-deviation scaling change weighting.

Control centering. Subtracting the shared unchanged-state control and then centering across the same candidate group cancels that control (Lemma 1). This algebraic identity motivates retaining the uncentered revision credit; it does not constitute an additional empirical recovery comparison.

D.4 Training and inference settings

Table 5 gives the shared training and inference settings.

The backbone is Qwen2.5-VL-7B-Instruct, with the same initialization and tokenizer across matched variants. RL starts directly from this checkpoint without additional supervised format warm-up; SFT is a separate comparison. AdamW updates the full language decoder while the vision encoder and multimodal projection remain frozen. Training uses BF16 and FSDP on four A100 80GB GPUs, with one start per GPU and four accumulation steps. The learning rate is constant without learning-rate warm-up; fresh rollouts receive one update epoch, with no replay. The final checkpoint after one data pass is retained, without validation-based checkpoint selection.

Both stages sum generated-token contributions. Rollout sampling uses temperature 1, top-$p = 1$ and no top-k filter; full support refers to the allowed tokens at each step, within the fixed length limits. The revision-stage weight is $\lambda_r = 1$, i.e., the unit revision/continuation weighting of Eq. (29), and $\eta = 0$; the citation weight $\alpha \in \{0, 1, 2\}$ is used only by the adapted citation baseline (Appendix H.2). Every training configuration is run with the same three seeds (Table 5). Inference uses greedy decoding at batch size 1 on one A100 80GB GPU, with one revision followed by three answers; the timings in Table 10 use this setting. Output budgets are reserved within the context limit; overlength starts are excluded before method comparison, and incomplete outputs follow the declared failure rule.

Table 5: **Configuration values.** A lists the evaluation protocol and inference settings; B lists the $\mathrm{RP}^2\mathrm{O}$ training settings. Matched baselines share the entries that apply to them.

A. Evaluation and inference

Parameter	Value
Random seeds	17, 42, 2026
Horizon H	3
Endpoint L	3
Revision decisions	1
Decoding	Greedy
Batch size (dialogues)	1
Hardware	1×A100 80GB
Context limit (tokens)	8192
Revision limit (tokens)	128
Answer limit (tokens)	256

B. Training

Parameter	Value
Learning rate	10^{-6}
Adam (β_1, β_2)	0.9, 0.999
Adam ϵ	10^{-8}
Weight decay	0.01
Data passes	1
Update epochs	1
Batch size (starts)	16
Accumulation steps	4
PPO clip	0.2
Gradient norm clip	1.0
KL coefficient	0.01
Candidate edits G	4
Continuations K	2
Control samples K_0	2
Revision-stage weight λ_r	1
Cost weight η	0
Sampling temperature	1

E Task Construction and Semantic Audit

E.1 Construction and data provenance

Construction begins with an image-level split and the source expressions, object annotations, and dialogue families. Each accepted task links observable task-defining information to a unique object and to a dependent query with an annotated target answer. Role-specific interventions then produce clean and corrupted estimates or explicit reselections. The constructed families are manually checked for referent identifiability and the dependence of their three follow-up questions on the maintained reference.

Training histories, shared diagnostic starts, and native benchmark inputs serve distinct purposes. Diagnostic histories are frozen before method comparison, independently of the compared methods' repair success. For a model-generated error pool, the generator, source prompt, sampling rule, and acceptance criteria are shared across all methods. Natural rollouts from each evaluated policy form a separate end-to-end condition.

Native benchmark runs preserve the released inputs, targets, and scoring rules. A common adapter may serialize information already present in those inputs; it cannot add an annotated intent or role only for $\mathrm{RP}^2\mathrm{O}$. Additional semantic checks and role-specific relabeling belong to the controlled diagnostic evaluation.

E.2 Role-specific task interventions

Let $\mathbf{a}^\star(x)$ be the target answers under a fixed subsequent-query protocol. For admissible replacement boxes b' and defining selections s', define

$$\begin{aligned} T^{\mathsf{E}}_{b'}x &:= (I, h, (d, s, b', \mathsf{E})), \qquad z = \mathsf{E}, \\ T^{\mathsf{D}}_{s',b'}x &:= (I, h, (d, s', b', \mathsf{D})), \qquad z = \mathsf{D}. \end{aligned} \tag{33}$$

These operators define task-construction interventions. Section 3.1 distinguishes revising a cached box from changing the task-defining selection. Appendix A gives the identifiability assumptions and multiple-reference extension.

Cached boxes. Shared starts use a correct box or a box around another plausible object. Cache-only changes preserve the intended object and downstream target answers; in particular, $\mathbf{a}^\star(T^{\mathsf{E}}_{b'}x) = \mathbf{a}^\star(x)$. Correct starts measure harmful revisions, whereas wrong-object starts test recovery.

User reselection. An explicit external selection s' sets the target to $\mathrm{Select}(I, s')$. The selection and cache are separate arguments; no map from s' to b' is assumed. Target answers are recomputed under the same subsequent-query protocol, although not every answer must change. Paired sensitivity tests use distinct validated targets. Within either task, the model's edit changes only b while keeping the task-defining information fixed.

Coherent histories. A coherent-error condition changes both the cache and model-origin claims so that they agree on an incorrect object, while retaining recoverable original task-defining information. It tests recovery when the model's history is internally consistent.

E.3 Sampling unit and accepted examples

The primary independent unit is an image with a dialogue family. Multiple role conditions and interventions within a family form paired observations. Image identifiers determine the split before dialogues, augmented histories, crops, or paraphrases are constructed. A global exclusion list handles shared MSCOCO images across datasets (Lin et al., 2014).

Each family records the originating description, immutable defining selection if present, parent identifiers, intended objects, target boxes, and the downstream task program or annotated target answer. Only observable fields enter normal policy inputs. Intermediate target boxes support auditing, evaluation, and supervised controls; the primary utility objective uses downstream answers. Verifier labels remain unavailable to the deployed policy. Correct reference boxes are supplied only in the separately declared execution diagnostic in Appendix G.1. Acceptance requires a unique intended object, recoverable originating information, a dependent answer sensitive to the reference, and an intervention consistent with its role.

E.4 Paired examples and leakage checks

A minimal pair shares the image and candidate box but differs in task definition: "continue with the person in blue; this is the cached estimate" versus "now use the person selected here." Neither wording identifies the cache as incorrect. Correct and incorrect estimates carry the same role tag. Reference identifiers are sequential, independent of segmentation IDs; candidate ordering is randomized and recorded.

E.5 Data sources and evaluation

Natural-image data. Native single-turn REC uses the RefCOCO-family images, expressions, and target annotations (Mao et al., 2016; Yu et al., 2016). Global MSCOCO-ID exclusions separate training and evaluation uses. REC applies the answer-stage policy to the original image and expression without injected dialogue history. RegionDial-Bench provides the unmodified multi-turn detection test (Sun et al., 2026); the constructed recovery tasks have their own role labels and follow-up questions. Native source-record counts and constructed recovery-family counts are distinct.

Reference-dependent follow-up questions. Each accepted continuation spans a fixed diagnostic horizon of $H = 3$ follow-up questions whose validated targets depend on the active reference, either directly or through recorded descendants. Dialogue position and dependency depth are distinct.

E.6 Diagnostic scale

Structural filtering retains 4012 distinct geometry links from 4281 candidates: 21 fail ancestry checks, 10 have identical reference and answer boxes, and 238 duplicate retained geometry. A deterministic image-level partition within these public benchmark data separates the shared evaluation cohort, an audit reserve and an unused pool. Table 6 gives their sizes and defines N as the number of links in the shared cohort. Each link defines one family, reused across roles and conditions. These structural counts are separate from the manual semantic checks described above.

Table 6: **Geometry statistics.** Counts after structural filtering. Partitions are image-disjoint. The shared cohort supplies the diagnostic starts for Tables 1, 2 and 7–10; the audit reserve and unused pool do not enter these comparisons.

Partition	Images	Geometry links
Shared cohort	750	1358
Audit reserve	200	349
Unused pool	1301	2305
Total	2251	4012

Held-out wording is paired with its explicit-role counterpart. Family counts, question counts and semantic pairs are different units; the same family can supply several paired conditions. The follow-up plots use the shared cohort in Table 6. Failed or missing model outputs do not reduce an eligible denominator.

E.7 Aggregation and reporting conventions

All diagnostic tables use $H = L = 3$ with exactly one fixed active reference. Each role contributes N sequences on the shared cohort. Endpoint rates use $100 \sum_n X_n/N$; Avg. Answer uses $100 \sum_{n,t} A_{nt}/(3N)$. JR uses $B_n A_{nL}$ and Pair requires joint success in both role variants of the same family. Correct-cache damage uses a separate start variant on the same N links. Means and sample SD use the three unrounded seed-level statistics, with $s^2 = \sum_{j=1}^{3}(x_j - \bar{x})^2/2$; SD is neither a standard error nor a family-cluster interval. Repeated follow-ups and seeds do not increase the number of independent image clusters. The image-cluster bootstrap intervals in Figures 2 and 4 concern variation over evaluation images, with the image as the resampling unit; they are distinct from the between-run variation summarized by table SDs.

No edit retains a wrong estimate cache and a correct defining cache, giving structural $\mathrm{R} = \mathrm{JR} = \mathrm{Pair} = \mathrm{Damage} = 0$ and $\mathrm{Def} = 100$; its answers can still be correct despite the wrong cache (Answer 16.96 in Table 1). Cache interventions supply each cached edit to both frozen executors, and conditions that coincide with a primary run reuse its records. An identical value across tables is therefore the same evaluation, not independent confirmation: the wrong-object row of Table 7A and the explicit-role columns of Table 7B repeat Table 1; the performance columns of Table 10 repeat Table 1; and the diagonal cells of Table 9 repeat Table 2.

E.8 MATCHED BRANCH ENVIRONMENT

For the controlled utility estimate, candidate and unchanged-state branches receive the same ordered follow-up questions, revealed one at a time. The query schedule is fixed before candidate sampling, independently of its success. Future queries, target boxes, and rewards are excluded from policy inputs. Branches differ only in the canonical active reference state and generated responses. The edited record remains fixed over the measured horizon, while the executor produces ordinary answers and task-required records. An adaptive-user extension requires a shared, explicit user-policy distribution to define comparable expected returns.

Coordinates are canonicalized by a fixed syntax rule. A cache edit updates all designated active aliases of the reference while preserving the originating user instruction. Historical model claims have the same historical-status marking in every branch. Coordinate-only and coherent-error conditions are reported separately.

F PROMPTS AND WORKED EXAMPLE

Prompts F.1 and F.2 give the complete system instructions for the example in Figure 3. One shared VLM makes one reference revision, then answers each revealed query with the submitted state. Blue cards denote revision, teal cards denote execution, and gray cards contain shared inputs.

Shared input | User-message template

```
IMAGE: <image>
HISTORY: {past_submitted_turns}
CURRENT_QUERY: {current_query}
READ_ONLY_REFERENCES: {observable_ancestor_records}
ACTIVE_REFERENCE:
  id: R1
  role: {ESTIMATE_or_DEFINITION}
  originating_description: {unchanged_original_expression}
  defining_selection: {explicit_selection_or_null}
  cached_box_xyxy: {normalized_xyxy_box}
  parents: {parent_reference_ids}
```

F.1 REFERENCE REVISION

Prompt F.1 | Reference revision | System

Task: Maintain the localization of the active visual reference.

Reference guidance: For ESTIMATE, the originating description and its context identify the intended object. For DEFINITION, the explicit defining selection identifies the intended object. The role does not indicate whether the cached box is correct.

Action: KEEP retains the cache. REBIND replaces only the cached box with a box for the same intended object. Preserve the task definition. Do not answer the current query in this stage.

Coordinates: Full-image normalized [x1,y1,x2,y2].
Require 0 <= x1 < x2 <= 1 and 0 <= y1 < y2 <= 1.

Format: Return one JSON record and no additional text:

```
{"id":"R1","action":"KEEP"}
or {"id":"R1","action":"REBIND","box":[x1,y1,x2,y2]}
```

F.2 DOWNSTREAM REASONING

Prompt F.2 | Downstream reasoning | System

Task: Answer the current object-localization query.

Reference guidance: Use the image, task definition, available history, and active reference state to resolve the question. For ESTIMATE, the original description establishes identity; for DEFINITION, the defining selection establishes identity. The requested answer object may differ from the reference object.

State: The active reference is read-only. Do not issue KEEP or REBIND, or overwrite R1 with the answer. The supplied box is not a correctness guarantee. Answer only the current query.

Coordinates: Full-image normalized [x1,y1,x2,y2].
Require 0 <= x1 < x2 <= 1 and 0 <= y1 < y2 <= 1.

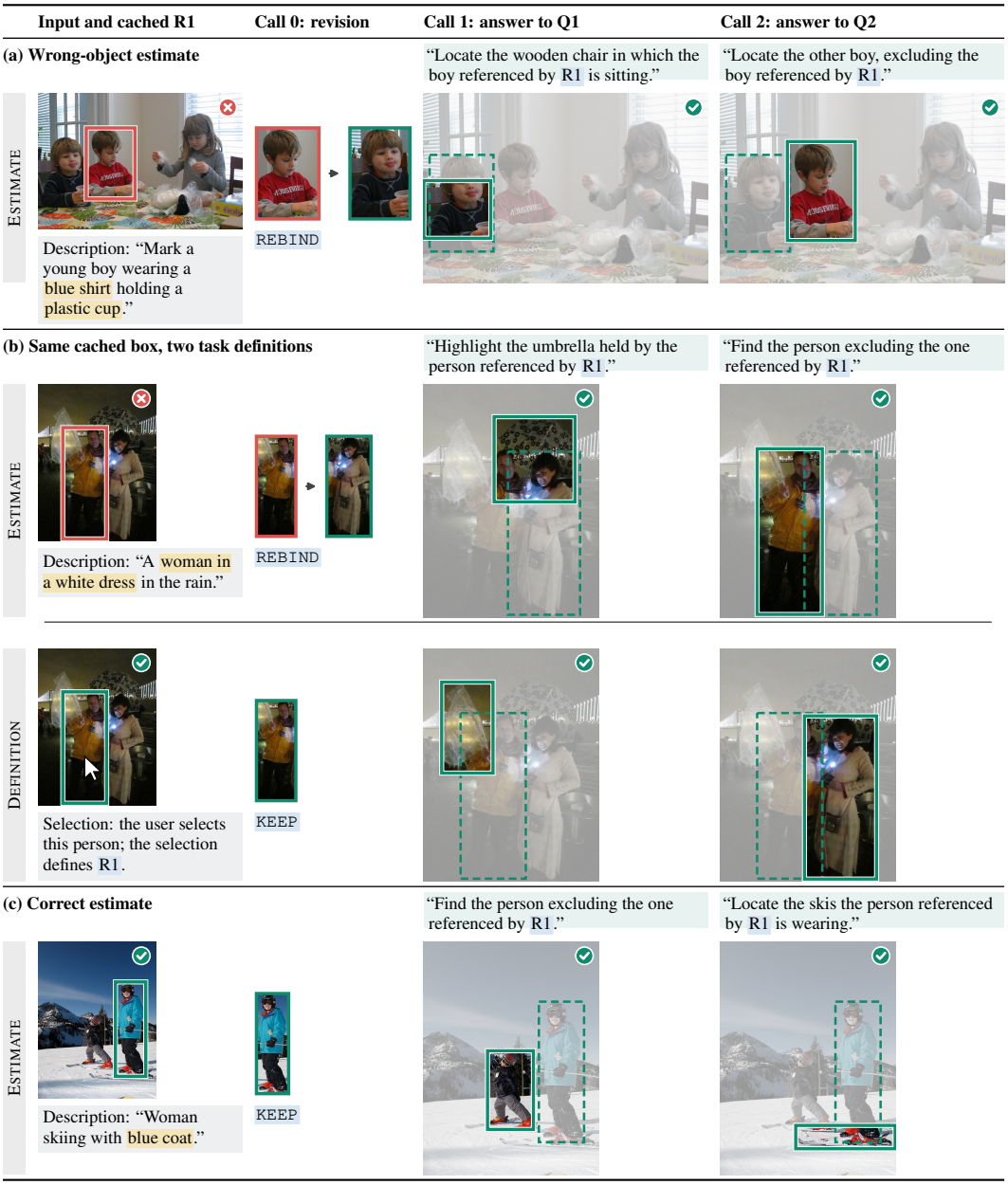
Format: Give a brief visual explanation and one target box:

```
<think>Brief visual explanation.</think>
<answer>{"box":[x1,y1,x2,y2]}</answer>
```

Message construction. Each call combines its stage-specific system instruction, the shared user template with that call's field values, and the unchanged source image. The executor uses the same instruction in the candidate and unchanged-state branches. It sees the resolved active state, not the revision action record.

F.3 WORKED EXAMPLE

Figure 3 illustrates three start conditions of the protocol. In (a), a wrong-object cache is followed by one `REBIND` to the boy specified by the original request; the same VLM then locates his wooden chair and, after the next query is revealed, the other boy. In (b), the same wrong cache is shown under both task definitions: the estimate is revised, whereas an explicit user selection of that person defines the target and is kept, so the same questions have different target answers. In (c), a correct estimate is kept. The call cards on page 28 give the inputs and outputs for (a).



Input: the box is the cached R1. Answers: solid box = output; dashed box = R1 as read by the call. Red / green = wrong / correct for the task. Text shading follows the call cards of Appendix F: input text (grey-blue), queries of Calls 1–2 (green), R1 and the Call-0 action (blue); yellow marks the words that identify the intended object.

Figure 3: **Examples of referent-preserving revision.** Columns show the shared input, the revision call (Call 0) and the first two follow-up queries (Calls 1–2) of the $H = 3$ protocol. In each answer the lit box is the output and the dashed box is the remembered reference R1. (a) The request identifies the boy in blue holding a cup, but the cache points to the boy in red; one `REBIND` moves R1 to the intended boy. (b) The same cached person under two task definitions: as an estimate of "a woman in a white dress" it is revised; as the user's explicit selection it defines the target and is kept, and the same questions resolve to different objects.

Concrete messages for Figure 3. The fixed fields and call-specific bindings below fully specify each user message through the shared template. In this excerpt, Prompt F.1 is used once; Prompt F.2 is reused twice. Q2 is revealed only after the answer to Q1 has been submitted.

Example bindings | Shared across calls

Image: the unchanged source photograph in Figure 3.

Originating description: Mark a young boy wearing a blue shirt holding a plastic cup.

Fixed fields: `id: R1; role: ESTIMATE; defining_selection: null; parents: []; READ_ONLY_REFERENCES: []`.

Call 0 | Reference revision

System: Prompt F.1. **History:** Empty (`[]`).

Current query: Locate the wooden chair in which the boy referenced by R1 is sitting.

Cached box: `[0.235938,0.255269,0.475000,0.777518]`.

Assistant output:

```
{"id":"R1","action":"REBIND","box":[0.020313,0.325527,0.245313,0.836066]}
```

Program step: Apply the action to the cache; pass the resulting state, not the action record, to Call 1. No model call occurs in this transition.

Call 1 | Answer Q1

System: Prompt F.2. **History:** Empty (`[]`).

Current query: Locate the wooden chair in which the boy referenced by R1 is sitting.

Cached box: `[0.020313,0.325527,0.245313,0.836066]`.

Assistant output:

```
<think>The chair is partly visible behind the boy holding
the cup on the left.</think>
<answer>{"box":[0.000000,0.461358,0.235938,0.765808]}</answer>
```

Call 2 | Answer Q2

System: Prompt F.2. **History:** The exact Q1 and complete assistant output from Call 1, serialized as prior user/assistant messages.

Current query: Locate the other boy, excluding the boy referenced by R1.

Cached box: `[0.020313,0.325527,0.245313,0.836066]`.

Assistant output:

```
<think>The other boy is the boy in red beside R1.</think>
<answer>{"box":[0.235938,0.255269,0.475000,0.777518]}</answer>
```

State after Call 2. R1 still denotes the boy in blue with cache `[0.020313,0.325527,0.245313,0.836066]`; the chair and other-boy boxes are separate task outputs.

Different task: explicit selection. If the user instead selects the boy in red as R1, use `DEFINITION` and set both defining selection and cache to `[0.235938,0.255269,0.475000,0.777518]`. The illustrative action is `{"id":"R1","action":"KEEP"}`.

F.4 Protocol and Information-Matched Controls

Action parsing and likelihoods. The KEEP command has one canonical serialization, denoted by u_0; REBIND replaces the cache while leaving the defining selection and other immutable fields fixed. The likelihood evaluates the sampled token record $u \in \mathcal{U}$. Distinct strings mapping to the same box remain distinct sampled outcomes unless the decoding grammar enforces unique serialization; Φ may be many-to-one. Replacement coordinates are finite, ordered, and in $[0, 1]$. Old and new likelihoods use identical syntax support. Deterministic parsing is part of Φ, with invalid actions handled by the declared failure rule. Teacher replacements and free retries are excluded from this sampling procedure.

Observation boundaries. The executor receives the original image, permitted history, immutable origin and role, and resolved active box after Φ. Its input excludes the revision action label, success flags, the control continuation, and uncommitted answers produced during revision. Provenance logs and version counters remain outside the prompt to avoid signaling whether an edit occurred. Archived model claims use the same historical-status convention in both branches. Evaluation records the submitted reference and task output.

The dependency graph supports auditing and evaluation; an autoregressive executor may still attend across branches. Required references are protocol-defined. The primary study makes one revision decision before a fixed continuation; a sequential extension requires explicit scheduling and likelihood accounting for later edits.

Re-grounding. This control resolves every active estimated reference before answering. Explicit selections continue to specify identity while permitting boundary refinement. The budget includes all additional model and image processing.

Wording transfer. The transfer condition replaces canonical role tags with checked natural-language task definitions for every compared method. Input fields and output parsers remain shared.

Native checkpoints. Native outputs are converted to the common evaluation fields when supported. Scores use task outputs and maintained references, with no requirement to reproduce $\mathrm{RP}^2\mathrm{O}$'s internal reasoning format. An extra query to infer an unreported reference counts as an additional model call. When native outputs do not support comparable joint-reference metrics, their native-schema scores provide task context. Native benchmark adapters use only released observable information; role annotations added during diagnostic construction are excluded from native runs.

G Recovery and Follow-up Evaluation

This section connects recovery, task preservation and subsequent answer utility. Table 7 tests error sources and wording; Figure 2 follows the consequences of one edit.

G.1 Evaluation conditions

Natural trajectories. Each method generates its own history under the same query schedule, then makes one revision decision at the protocol-fixed opportunity. Table 7C reports initial active-cache correctness and subsequent Avg. Answer and JR on all eligible families, without selecting only erroneous starts. Missing histories or outputs fail under the common parser protocol and remain in the denominator. These results include the errors each policy encounters; a model with fewer initial errors need not be better at repairing a fixed error.

Shared starting states. Every method receives the same frozen image, history, task-defining information and reference cache. Common correct and corrupted sets support paired estimates of damage and recovery. Controlled corruptions and coherent model-generated errors retain separate source labels. Pool construction and acceptance precede inspection of compared methods' recovery outcomes (Appendix E). The no-edit replay uses the same executor as the edited branch.

Ground-truth-reference execution. The diagnostic supplies the annotated active cache once, at the same intervention point as the learned edit. Its no-edit control retains the corrupted cache; neither condition runs an editor. The assigned cache remains fixed through the continuation. Other history

fields and any subsequently generated references follow the common executor protocol; they are not replaced by annotations at later turns. Historical claims keep the same declared status, and no correction flag, answer label or future query is exposed. Table 9 keeps this input diagnostic separate from learned edits and deployment. Correct inputs do not guarantee an upper bound.

G.2 Evaluation metrics

For the single-active-reference diagnostics reported here, $B_{nt} = B_n$ indicates whether the fixed active cache in family n identifies its intended object, and A_{nt} indicates whether the answer at question t is correct. Submitted answer boxes are scored through A_{nt}; neither they nor subsequently generated references enlarge the reference set defining B_n. This cache-level indicator is the instance-level counterpart of C_t in Eq. (3). For reference correctness, the intended object must be the unique maximum-IoU match with IoU at least 0.5 in the identity-audited box subset. Ties are resolved or excluded before model outcomes are inspected.

For a submitted answer box $\widehat{a}_{nt}$ and its annotated target box $a^\star_{nt}$, the answer verifier is

$$A_{nt} = \mathbb{1}[\widehat{a}_{nt} \text{ is valid and } \mathrm{IoU}(\widehat{a}_{nt}, a^\star_{nt}) \geq 0.5]\,. \tag{34}$$

Missing or invalid answer outputs score zero. The answer target can differ from the active-reference object. For the reported answer-utility runs, $v(y_t, a^\star_t(x)) = A_{nt}$, so $U = H^{-1}\sum_t A_{nt}$ at $\eta = 0$. Reference correctness is evaluated separately and is not multiplied into the training answer reward. Define $S_{nt} = B_{nt}A_{nt}$.

On a fixed common set $\mathcal{E}$ of recoverable corrupted estimates, and paired scene families $\mathcal{P}$, use

$$\mathrm{JR@}\,L = \frac{1}{|\mathcal{E}|}\sum_{n\in\mathcal{E}} S_{n,L}, \qquad \mathrm{Pair} = \frac{1}{|\mathcal{P}|}\sum_{n\in\mathcal{P}} S^{\mathsf{E}}_{n,L}S^{\mathsf{D}}_{n,L}. \tag{35}$$

Reference recovery and endpoint answer accuracy are $\mathrm{R@}\,L = |\mathcal{E}|^{-1}\sum_n B_{n,L}$ and $\mathrm{Answer@}\,L = |\mathcal{E}|^{-1}\sum_n A_{n,L}$ on this same set. The main diagnostic sets $L = H$ after a fixed $H \geq 2$ follow-up questions. K denotes sampled continuations, not dialogue length. At $\eta = 0$, the empirical answer utility is

$$\overline{\mathrm{Answer}} = \frac{1}{|\mathcal{E}|}\sum_{n\in\mathcal{E}}\frac{1}{H}\sum_{t=1}^{H} A_{nt}. \tag{36}$$

Tables report this as Avg. Answer in percent; Eq. (6) uses the corresponding fraction. All scheduled turns count, with missing outputs scored zero. A cost-sensitive study would report answer correctness and resource cost separately as well as U.

On a shared initially correct set $\mathcal{C}$, Damage is the fraction whose submitted reference is incorrect. Definition compliance is the fraction of definition-role tasks whose active caches identify the selected objects at L. These sets are distinct from $\mathcal{E}$. A valid edit keeps the task definition fixed but can still yield an incorrect localization.

The four reference–answer outcomes on $\mathcal{E}$ have proportions JR, R$-$JR, Answer$-$JR and $1-$R$-$Answer$+$JR, with scores expressed as fractions. This decomposition explains the answer–joint gap without supplying an independent result. It differs from before–after reference transitions, and cannot by itself identify a bypass mechanism. Uncertainty is computed from joint per-example and per-seed records, not reconstructed from marginal SDs.

G.3 One edit and subsequent questions

Figure 2 compares retained caches with edits from GRPO (Answer), GRPO (IoU) and $\mathrm{RP}^2\mathrm{O}$, using a frozen GRPO (Answer) executor. All branches share the image, history, intended object, schedule and decoding. Edits are cached before continuation evaluation; the executor receives only the canonical state. Figure 2(b, c) reports Answer@t and JR@t with image-cluster intervals. Relative to the retained cache, $\mathrm{RP}^2\mathrm{O}$ edits raise Answer@t by 38.69, 35.81 and 38.42 points at $t = 1, 2, 3$ (37.64 on average), compared with average gains of 33.72 for GRPO (Answer) edits and 33.27 for GRPO (IoU) edits; because all branches use the same fixed cohort, these differences of means equal the mean paired differences. Under a correct annotated cache, JR@t equals Answer@t by construction,

and under the retained corrupted cache JR@t is structurally zero. These input conditions provide comparison points, not guaranteed bounds on answer accuracy. The JR@t values in Figure 2(c) are therefore also the paired JR gains over the retained cache (75.72, 71.67 and 72.43 for $\mathrm{RP}^2\mathrm{O}$). Table 9 checks whether the interpretation changes with a second executor.

The horizon is fixed before test evaluation. A common cohort with audited questions at every position supports within-family comparisons; its coverage and exclusions are recorded. Dialogue position, elapsed distance from the edit, and dependency depth are separate fields. A later question must actually depend on the maintained reference, rather than independently repeating its full target description. Where new downstream references are generated, their identifiers, target labels and dependency paths are recorded.

Because the scored active cache is fixed, its correctness cannot change merely because another question was answered. Accordingly, the follow-up plots measure answer and joint correctness, not repeated repair decisions. Differences between later and earlier questions can reflect task difficulty; edit effects are assessed through paired branch comparisons at each question, not through the curve's slope alone. The common-start study does not establish adaptive-dialogue or repeated-revision performance.

G.4 Error sources and held-out wording

Table 7A focuses on wrong-object caches, coherent erroneous histories and a frozen pool of natural model errors. Each family contributes one endpoint for each declared condition; n counts eligible families including failed or missing submissions. Natural shared-start errors remain distinct from each method's own trajectories.

Panel B uses paired explicit-role and held-out natural-language descriptions of the same audited tasks. JR is evaluated on estimate-role starts and Pair on complete semantic pairs. Correct and incorrect estimates share the same user query. Splits are disjoint by wording family as well as by image where required. Panel C uses the full fixed family set with model-specific histories; its initial-reference fraction concerns the active cache before the edit. It is not subtracted from Panel A as a causal estimate.

$\mathrm{RP}^2\mathrm{O}$ has the highest JR under every error source in Panel A and the highest JR and Pair under both wordings in Panel B. Coherent histories are the hardest source for all three methods (58.25, 56.63 and 66.74 JR). In Panel C, GRPO (IoU) starts with the most correct active references (79.68) but has the lowest Avg. Answer and JR@L (78.28 and 75.97), so initial correctness alone does not determine the endpoint.

G.5 Baselines and information budget

Native Qwen2.5-VL, VisionReasoner and RegionReasoner provide task context; their different training histories prevent causal attribution (Bai et al., 2025; Liu et al., 2026b; Sun et al., 2026). Same-initialization comparisons share input fields, annotation access, trainable parameters and precision. RL variants use no additional supervised warm-up. Training comparisons account for all generated rollout tokens. SFT is a separate supervised comparison using reference-correction labels from the resolved-box annotations available to the IoU comparator and auditing.

Always re-grounding resolves the active reference from its original task-defining information; calls and generated tokens are included in cost. The no-edit row replays the $\mathrm{RP}^2\mathrm{O}$ executor with the original cache. These comparisons distinguish retaining the cache from unconditional re-localization and learned revision.

The RegionReasoner-style citation row is a reward adaptation on the common backbone and recovery interface, not the native checkpoint's recovery score. Removing citation reward changes only that adapted condition; its zero-citation setting is not identified with the GRPO (Answer) row. The citation-weight contrast is interpreted within the adapted baseline, rather than as an explanation of its absolute performance gap from GRPO (Answer). Besides lowering R, JR and Damage (Section 4), the citation term raises definition compliance from 85.81 to 93.20, consistent with retaining defining caches that start correct.

Table 7: **Recovery across error sources and wording.** Means ± sample SD (%) over three seeds; all conditions reuse the cohort in Table 6. A varies the error source on shared starts. B compares explicit role tags with held-out natural-language wording of the same tasks. C uses each method's own histories and keeps every family. The wrong-object row of A and the explicit-role columns of B repeat Table 1.

A. Error sources: JR (%) ↑

Error source	GRPO (Answer)	GRPO (IoU)	$\mathrm{RP}^2\mathrm{O}$
Wrong-object	$66.05_{\pm 1.03}$	$63.72_{\pm 1.30}$	$72.58_{\pm 0.52}$
Coherent	$58.25_{\pm 0.77}$	$56.63_{\pm 1.88}$	$66.74_{\pm 0.97}$
Natural	$62.20_{\pm 0.63}$	$60.26_{\pm 1.38}$	$69.59_{\pm 0.71}$

B. Explicit roles and held-out wording (%)

	JR ↑		Pair ↑	
Method	Explicit	Held-out	Explicit	Held-out
SFT	$56.41_{\pm 0.59}$	$51.01_{\pm 0.43}$	$47.74_{\pm 0.63}$	$42.29_{\pm 0.79}$
GRPO (Answer)	$66.05_{\pm 1.03}$	$61.93_{\pm 0.78}$	$56.73_{\pm 1.29}$	$52.95_{\pm 0.92}$
$\mathrm{RP}^2\mathrm{O}$ (Ours)	$72.58_{\pm 0.52}$	$69.24_{\pm 0.77}$	$65.68_{\pm 1.10}$	$62.13_{\pm 1.45}$

C. Method-specific histories (%)

Method	Initial active reference correct	Avg. Answer	JR@L
GRPO (Answer)	$78.65_{\pm 1.58}$	$85.88_{\pm 0.11}$	$82.28_{\pm 0.22}$
GRPO (IoU)	$79.68_{\pm 1.66}$	$78.28_{\pm 0.50}$	$75.97_{\pm 0.55}$
$\mathrm{RP}^2\mathrm{O}$ (Ours)	$75.06_{\pm 1.06}$	$85.31_{\pm 0.51}$	$84.05_{\pm 0.63}$

GRPO (IoU) uses ordinary GRPO with the cache-and-answer utility in Eq. (37); Table 2A uses the same utility within $\mathrm{RP}^2\mathrm{O}$. Panel B varies credit at fixed task utility and interface, with generated-token comparisons in Appendix H.4. GRPO (Answer) uses the mean binary correctness of the three follow-up answers as a shared terminal return for the edit and continuation. It shares a row with terminal return-to-go under terminal-only rewards, unit discount, and identical advantage transformations and update weighting. The ablation without revision control replaces $A_i^r = \widehat{Q}_i - \widehat{\widehat{Q}}_0$ with $A_i^r = \widehat{Q}_i$, while retaining the editor, within-edit execution credit in Eq. (10), advantage scales and token weights. Revision control and execution control denote $\widehat{Q}_0$ and the within-state leave-one-out reference term. Identical configurations reuse runs across tables (Appendix E.7).

G.6 Native benchmark reporting and scope

We follow the RegionReasoner protocol (Sun et al., 2026). RegionDial detection AP uses the native scorer and aggregation; REC reports Acc@0.5. Native adapters add no annotated intent or role unavailable to competing methods. The controlled recovery evaluation is reported separately from these native benchmarks.

H Mechanism and Optimization Analysis

The controlled analyses distinguish citation incentives, cache editing and execution, and the cost of learning.

H.1 Why the compared methods revise differently

Figure 1 contrasts retaining a wrong estimate with revising it. Several trained baselines also revise; they differ in what their training signal credits an edit for. Table 8 collects the intervention and utility definitions used in these comparisons.

Table 8: **Baseline interventions and training signals.** SFT is a separate supervised comparison. The citation contrast varies a reward component within its adapted condition; it does not compare a native RegionReasoner checkpoint with GRPO (Answer).

Condition	Training signal or evaluation intervention
No edit	Retain the original cache and run the $\mathrm{RP}^2\mathrm{O}$ executor.
Re-grounding	Re-localize unconditionally from the original task-defining information once, before the three answers.
SFT	Supervised reference correction using resolved-box annotations.
Citation adaptation	Common-backbone reward adaptation; vary the literal-citation weight while holding the remaining adapted condition fixed.
GRPO (IoU)	Ordinary GRPO with the equally weighted cache-and-three-answer IoU utility in Eq. (37).
GRPO (Answer)	Ordinary GRPO with mean binary follow-up correctness as the shared terminal utility for the edit and continuation.
$\mathrm{RP}^2\mathrm{O}$	Answer utility with separate edit and execution credits, Eqs. (9)–(10).
w/o revision control	Use $A_i^r = \widehat{Q}_i$; retain the same editor and within-edit execution credit.

How the model detects an error. The policy receives no correctness flag, and role cues appear on correct and corrupted starts alike (Appendix D). A revision is possible because the originating description or defining selection remains in the input next to the cached box. The low damage on correct starts (3.76) indicates that $\mathrm{RP}^2\mathrm{O}$ does not revise indiscriminately.

Answer reward and reference recovery. Under GRPO (Answer), an edit that leaves the answers unchanged receives the same terminal reward whether or not it repairs the reference. On corrupted caches, its frozen executor answers correctly more often than the $\mathrm{RP}^2\mathrm{O}$ executor (42.96 vs. 17.34 Avg. Answer; Table 9). Holding either executor fixed, $\mathrm{RP}^2\mathrm{O}$ edits raise JR over GRPO (Answer) edits by 6.2–6.4 points, so the difference is present in the edited caches. The cache interventions show different behavioral sensitivity to the supplied reference; they do not identify the models' internal reasoning mechanisms. The Answer–JR gap separately quantifies correct answers accompanied by an incorrect maintained reference.

Which components matter. Replacing downstream utility with IoU utility lowers JR by 4.88 points and Pair by 7.33 (Table 2A). Removing the unchanged-state control lowers JR by 1.18 points and slightly raises Avg. Answer (Table 2B). These comparisons show different recovery outcomes under different reward and credit configurations; they do not separately identify how much of a difference arises from revision learning and execution learning in the shared model.

H.2 Citation incentives and recovery

The mechanism test varies literal-citation weight $\alpha \in \{0, 1, 2\}$ in an otherwise fixed adapted baseline, with common clean and corrupted starts. The interpretation depends on citation extraction and reward timing. In particular, citing an old box need not prevent correcting the localization; the binary calculation in Appendix B.2 does not rule out such paths.

Figure 4 reports p_{ent} and p_{rec} separately, with image-cluster bootstrap intervals, together with the `REBIND` rate on corrupted and correct starts. Wrong-object caches, coherent erroneous histories and natural-error starts retain separate source labels. The action rate distinguishes two readings of a falling p_{rec}: a shift toward `KEEP` on both start types, which is the mechanism of Proposition 1, or a constant `REBIND` rate with more incorrect replacement boxes, which would indicate degraded localization instead.

From $\alpha = 0$ to $\alpha = 2$, the pooled `REBIND` rate falls from 69.6 to 38.3 on corrupted starts and from 13.2 to 4.1 on correct starts. Error entry falls for both roles (estimate 8.2 to 1.7; definition 14.2 to 4.3), and recovery falls under every error source (wrong-object 48.9 to 28.9, natural 44.7 to 26.3, coherent 42.3 to 23.8). These trends are consistent with increased retention of cached references, as in the binary model. The pooled action rates do not establish that replacement accuracy is unchanged within each error source, nor do they identify a unique explanation for the recovery decrease. The

finding concerns this adapted citation reward and does not imply that every historical-citation scheme prevents recovery.

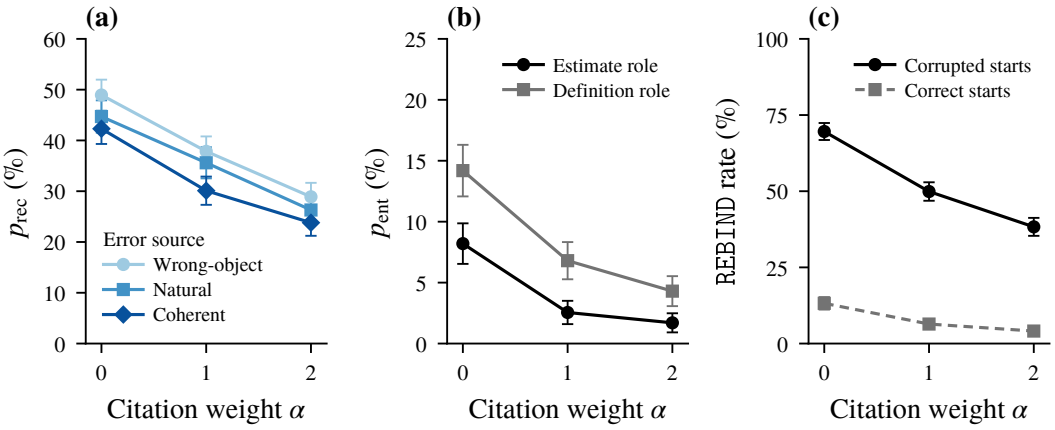


Figure 4: **Citation incentives and recovery.** The literal-citation weight α of the adapted RegionReasoner-style baseline is varied on shared starting states. (a) Recovery p_{rec} (R on corrupted starts) by error source: wrong-object caches ($n = 1358$), natural model errors and coherent erroneous histories. (b) Error entry p_{ent} on correct starts: for the estimate role it equals Damage, for the definition role $100-$Def. (c) REBIND rate on corrupted starts, pooled over the three error sources in (a), and on correct starts, pooled over the two roles in (b); this action underlies both panels. Error bars are 95% image-cluster bootstrap intervals. The $\alpha = 0$ and $\alpha = 1$ points of (a) wrong-object and of (b) are the citation rows of Table 1.

H.3 Cache editing and downstream execution

Table 2A holds interface, initial training checkpoint, two-stage estimator and sampling allocation fixed while changing utility. Let b_i be the active cache after edit u_i, $b^\star$ its annotated reference target, and $\widehat{a}_{ikt}$ the submitted answer box at follow-up t in continuation k. Write $q(b, b^\star)$ for continuous IoU, with $q = 0$ for a missing or invalid box. The IoU utility is

$$U_{ik}^{\text{IoU}} = \frac{q(b_i, b^\star) + \sum_{t=1}^{H} q(\widehat{a}_{ikt}, a_t^\star)}{H + 1}, \qquad H = 3. \tag{37}$$

The active cache contributes once, and each of the three answer boxes contributes once, all with weight $1/4$. Each answer is matched to its own annotated target, which may differ from the reference target. The denominator stays four when outputs are missing or invalid; an invalid edit has zero utility under the failure-state rule. The unchanged-state control uses the same formula with the original cache and its own continuations. This continuous IoU utility differs from the binary answer verifier in Eq. (34); both utility conditions have the same annotation access and $\eta = 0$.

Within a fixed valid edit, $q(b_i, b^\star)$ is constant across continuations and cancels from the leave-one-out execution advantage. The answer-box terms remain, so execution credit need not vanish. This comparison changes reference supervision, continuous versus binary answer scoring, and their relative weighting; it does not isolate revision learning from execution learning in the shared model. Avg. Answer measures the binary follow-up objective; JR, Pair and Damage describe reference outcomes and tradeoffs. Endpoint JR alone cannot establish an increase in mean answer utility.

Revision and execution share parameters. In Table 9, each editor's output is cached once and supplied to both frozen executors; each executor also receives retained and annotated caches on the same families. Frozen means that executor parameters are held fixed across the supplied-cache conditions. The active cache is set at the common intervention point and remains fixed thereafter. Continuations are evaluated after fixing the supplied cache, with the same decoding rule in every branch. Holding the executor fixed tests edit effects; holding supplied caches fixed tests execution effects. Disabling one training loss in a shared model would not freeze the other component.

Table 9: **Cache interventions under frozen executors.** Cells give Avg. Answer / JR@L (%), each with sample SD over three seeds on the shared cohort in Table 6. The active cache is set once. Each executor uses the same question schedule and decoding across supplied caches; learned diagonals and the no-edit RP^{2}O condition reuse the runs in Tables 1–2. Ground truth denotes an annotated input box; method rows name the editor. Corrupted caches have structurally zero JR, shown as a dash.

	Frozen executor	
Box source	GRPO (Answer)	RP^{2}O
Corrupted	$42.96_{\pm 1.06}$ / -	$17.34_{\pm 1.13}$ / -
Ground truth	$92.22_{\pm 0.10}$ / $91.92_{\pm 0.36}$	$92.70_{\pm 0.21}$ / $92.69_{\pm 0.41}$
GRPO (Answer)	$76.68_{\pm 0.55}$ / $66.05_{\pm 1.03}$	$70.50_{\pm 0.69}$ / $66.42_{\pm 0.85}$
RP^{2}O (Ours)	$80.60_{\pm 0.36}$ / $72.43_{\pm 0.49}$	$75.81_{\pm 0.61}$ / $72.58_{\pm 0.52}$

Figure 2 gives the per-question view of the same type of intervention. Its RP^{2}O and GRPO (Answer) cells reuse the corresponding cached edits when starts and decoding coincide. The native full-model endpoint in Table 1 remains distinct from a cross-checkpoint diagnostic. The diagonal cells of Table 9 coincide with the primary runs (76.68/66.05 and 75.81/72.58 in Table 2).

H.4 Credit assignment and generated-token budgets

The full RP^{2}O configuration uses G edit generations and $GK + K_0$ continuations per start, with $(G, K, K_0) = (4, 2, 2)$. Figure 5 plots held-out Avg. Answer and JR@L against cumulative generated rollout tokens, including every candidate and unchanged-state branch. Token accounting covers every branch actually generated, including any control samples that do not contribute a training loss. Removing a term from the advantage therefore does not by itself remove its generated tokens from the budget. This comparison measures generated-token budgets rather than total compute.

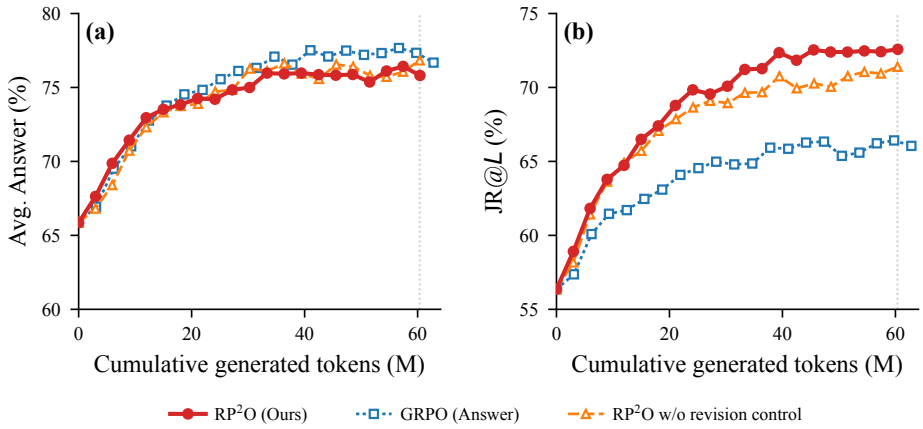


Figure 5: **Credit assignment and generated-token budgets.** Held-out (a) Avg. Answer and (b) JR@L versus cumulative generated rollout tokens, including all candidate and unchanged-state control branches. Checkpoints occur at fixed fractions of a single data pass and are plotted at their measured token counts, yielding slightly different final budgets. The dotted line marks the largest budget reached by every variant. Lines show means over three seeds.

The mechanism study uses $\eta = 0$, and the citation weight α is distinct from both the KL coefficient β and the revision-stage weight λ_r.

I Inference Cost

Timing includes all calls, image processing and prefill under shared hardware, batching and decoding.

Table 10: **Performance and inference cost.** Means ± sample SD over three seeds; performance (%) reuses Table 1. Costs are per dialogue and cover all three answers and one edit when enabled; calls are fixed by the protocol. For No edit, dashes denote structural zeros in JR and Damage.

Method	Answer ↑	JR ↑	Damage ↓	Calls	Generated tokens	Time (s)
No edit	$16.96_{\pm 0.68}$	-	-	3	$287.39_{\pm 0.25}$	$7.65_{\pm 0.01}$
Re-grounding	$62.44_{\pm 0.32}$	$55.15_{\pm 0.60}$	$13.11_{\pm 1.68}$	4	$309.42_{\pm 0.25}$	$8.36_{\pm 0.01}$
$\mathbf{RP^2O}$ (Ours)	$75.68_{\pm 0.53}$	$72.58_{\pm 0.52}$	$3.76_{\pm 0.60}$	4	$307.35_{\pm 0.31}$	$8.30_{\pm 0.01}$