WHEN SHOULD A VLM LOOK? PAYING ONLY FOR VISUAL CALLS THAT WERE NEEDED AND USED

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language agents that crop and zoom are trained with rewards that credit a successful tool call, yet a successful call does not show that the model needed to look or used the pixels it received. On our cold-start checkpoint only 10% to 12% of visual calls were both needed and used, and released agents make spurious calls 36% to 87% of the time on individual benchmarks. Outcome rewards, judge rewards, and branch probes each observe one side of this failure, and about two thirds of what an outcome reward pays goes to calls that were neither needed nor used. **CounterCredit** asks both questions of every image-returning call at its realized pre-call state, using the policy's own gold-answer score. A decision value compares the realized visual branch with answering immediately; an evidence value compares the returned crop with random same-size patches substituted into the same call. A call verified on both earns cashback and every other executed call pays rent; the price is bounded so that every correct trajectory outranks every wrong one, and a dual-channel GRPO advantage keeps the price in its own units. From the same cold start, prompt pool, and budget, **CounterCredit** reaches 89.5% on V*, 80.2% on HR-Bench-4K, and 76.4% on HR-Bench-8K, 6.3 to 9.4 points above outcome-only GRPO at 1.78 against 1.84 calls per question, and lowers the spurious-call rate to 31% to 36%, the lowest among the agents evaluated. The same recipe lifts a Qwen3-VL-8B base from 75.4 to 80.8 on average.

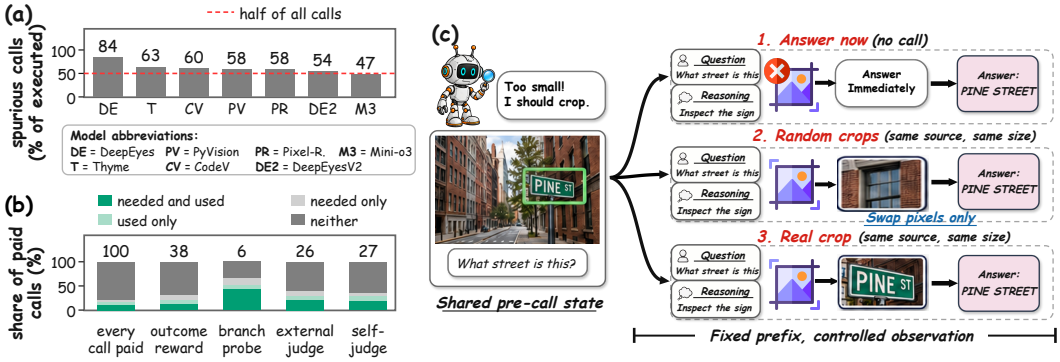


Figure 1: **Overview of spurious visual calls.** (a) Spurious-call rates of released visual agents. (b) Share of calls paid by different reward schemes on the cold-start checkpoint, split by whether they were needed and used; the number above each bar is the calls paid per 100 executed. Both panels average V*, HR-Bench-4K, and HR-Bench-8K. (c) Fixed-prefix counterfactuals that separate call necessity from evidence use while holding the realized pre-call state fixed across continuations.

1 INTRODUCTION

Multimodal models have made remarkable progress in visual understanding, yet VLMs still lag text-only language models as agents that reason through tools and acquire information on their own (Liu et al., 2025a; Wu et al., 2026; Zhang et al., 2026a; Zheng et al., 2026; Sathish et al., 2026). One possible bottleneck is that pretrained visual knowledge is not fully unlocked during

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.

post-training (Xiao et al., 2026): a VLM may understand an image when it is given one, yet fail to acquire visual information when it must decide what to inspect (Zhu et al., 2026). Tool use offers a natural mechanism to bridge this gap: an agent can inspect image regions with a crop tool. Yet trajectory-level success does not reveal whether a visual interaction was useful (Chen et al., 2026).

Our key observation is that a successful visual tool call does not show that the decision depended on the look or on the pixels it returned. Conventional outcome-based evaluation observes only final correctness and therefore cannot distinguish genuinely useful visual interactions from calls that were unnecessary or merely performative (Hou et al., 2026; Chng et al., 2026; Zhu et al., 2026; Lai et al., 2026). We refer to calls exhibiting either failure as *spurious*. As illustrated in Figure 1, this problem is widespread: averaged over three benchmarks, 47% to 84% of the calls made by released visual agents are spurious, and roughly two thirds of the credit assigned by outcome-based rewards goes to calls that are neither needed nor used. Judge-based rewards assess whether an observation appears useful (Wang et al., 2026d), while efficiency penalties merely discourage excess calls (Wang et al., 2026a); neither establishes that the call was necessary or that its evidence changed the decision. Existing objectives thus conflate *decision value* with *evidence value*, effectively paying agents for looking without verifying that they have seen (Zhao et al., 2026b).

We introduce **CounterCredit**, a counterfactual framework for assigning credit to visual interactions. For each visual call, we fix the pre-call prefix and score three continuations, answering without the call, the call with a matched random observation, and the call with the real observation, which separately measure whether the call is necessary and whether its visual evidence affects the decision. We use these quantities to construct a cashback-and-rent reward: a call on a correct trajectory earns cashback only when looking improved the answer and the improvement came through its returned pixels, and every other executed call pays rent. We then optimize this reward with a dual-channel GRPO (Shao et al., 2024) objective that normalizes the outcome and the price in separate channels, so that the price keeps its own scale against the outcome.

Contributions.

1. We identify *spurious visual calls* as a possible bottleneck for multimodal agentic reasoning, introduce two counterfactual values that characterize four types of visual interactions, and show that existing outcome-based rewards assign much of their credit to calls with little or no decision benefit, even with trajectories that reach the correct final answer.
2. We develop **CounterCredit**, a cashback-and-rent credit mechanism with a dual-channel GRPO objective that pays verified evidence use and charges unproductive calls, with an outcome-dominance guarantee that preserves the ordering induced by correctness.
3. CounterCredit averages 78.5 on the four perception benchmarks, 2.5 points above Mini-o3, the strongest tool agent on these four, and has the lowest spurious-call rate on V*, HR-Bench-4K, and HR-Bench-8K, 31% to 36%, 9 to 15 points below Mini-o3. It leads every tool agent on all seven benchmarks and remains effective with the Qwen3-VL-8B backbone.

2 Related Work

Learning visual tool use. V* studies guided visual search (Wu & Xie, 2024); DeepEyes learns interleaved visual reasoning through RL (Zheng et al., 2026), Pixel Reasoner adds curiosity rewards (Su et al., 2025), Thyme enables executable image processing (Zhang et al., 2026b), and Mini-o3, our cold-start source and baseline, trains long multi-turn search with an outcome-only reward from an LLM judge (Lai et al., 2026). VTool-R1 (Wu et al., 2026), ACTIVE-o3 (Zhu et al., 2026), and Chain-of-Focus (Zhang et al., 2025a) train the zoom decision itself, each rewarding the outcome of the trajectory that contains it. Outcome success alone does not establish evidence use.

Rewards for visual actions. CodeV's TAPO judges tool-output relevance (Hou et al., 2026); FaithEyes uses self-judging to score process images (Wang et al., 2026c). SCCM rewards externally judged sufficiency of the evidence (Liu et al., 2025b), and ViRL shapes step rewards from ground-truth visual rationales (Wang et al., 2025a). AdaTooler-V scales a tool-use reward component by a per-sample tool benefit score estimated from accuracy with and without the tool (Wang et al., 2026a). TACO probes answers before and after a tool branch (Feng et al., 2026), AXPO resamples the call and

its continuation from a fixed thinking prefix and scores the branch by its outcome (Kang et al., 2026), and IGPO rewards sequential belief updates (Wang et al., 2026b). TACO and the tool benefit score measure whether the branch helped, and CauAudit measures whether the pixels mattered but only as a post hoc diagnostic, so none measures both at the same realized pre-call state during training.

Credit assignment and causal attribution. PACR rewards increases in gold-answer probability along reasoning traces (Yoon et al., 2025); GiGPO estimates step advantages by grouping repeated environment states (Feng et al., 2025). Ma et al. (2026) analyze training effects with a no-tool RL control, finding that training mainly reduces tool-induced harm. Post hoc audits find the same pattern: most questions a tool agent answers need no tool (Guo et al., 2026), a placeholder in place of the returned image leaves aggregate accuracy intact (Shao et al., 2026), and models often ignore the evidence they fetch (Zhang, 2026). Counterfactual explanation tests (Turpin et al., 2023) and causal mediation (Pearl, 2001) provide related foundations; CauAudit diagnoses visual-evidence dependence through interventions (Wang et al., 2026e). CounterCredit brings such interventions into training and asks at the same pre-call state whether the look was needed and whether the pixels were used.

3 COUNTERCREDIT

A successful visual call leaves two questions unresolved: **was the look needed, and were the returned pixels used?** CounterCredit answers both at the realized pre-call state, Figure 2. *Decision value* D_i is the gold-answer score gain of the realized branch over answering at once, the total value of a look; *evidence value* E_i is its gain over random patches substituted into the same call, the part carried by the pixels. A call verified on both earns cashback; every other executed call pays rent.

3.1 SETTING AND THE COUNTERFACTUAL PROBE

Let $x = (I, y)$ denote an example with image I and gold answer y; the prompt includes the question and, for multiple-choice questions, the options the benchmark supplies, given verbatim in Appendix E. Policy π_θ generates a trajectory τ interleaving text with $n(\tau)$ executed visual calls $c_1, \ldots, c_{n(\tau)}$. The policy has one visual tool, a crop of the image. Each call c_i returns an image crop o_i, and the final answer receives outcome reward $r_{\text{out}}(\tau) \in \{0, 1\}$.

Let h_i be the exact prefix immediately before c_i, including the prompt, previous interactions, and generated reasoning; Appendix E gives the turn structure and Appendix F.7 four trajectories. Each probed call thus separates the pre-call context h_i, the call text c_i, and the returned pixels o_i; all three continuations below retain h_i.

All counterfactuals begin from the same realized h_i. The tool is a crop, and only a crop, because the evidence value needs a matched null return: a crop returns only pixels from I, so random same-size patches from I serve as that reference, while a tool that returns anything else has no such reference yet. We compare three continuations of h_i, shown in Figure 2(a): the answer-now continuation elicits an answer directly from h_i; the real-observation continuation (h_i, c_i, o_i) appends the call the policy actually issued and the crop it received; and each of K random-observation continuations $(h_i, c_i, \tilde{o}_{ik})$ appends the same c_i and an unscreened same-size patch sampled uniformly at random from the original image and rendered in the same format as the real observation.

For each of these prefixes p, define the policy's gold-answer score as

$$U_\theta(p) = \frac{\pi_\theta(y \mid p, e)}{\sum_{a \in \mathcal{O}} \pi_\theta(a \mid p, e)}, \tag{1}$$

where e is a fixed answer elicitation appended as a user turn, given verbatim in Appendix E, and $\mathcal{O}$ is the set of option letters of the question, so the score renormalizes the gold-option probability over them. For free-form tasks, we instead use the mean log-probability per token of the reference answer y under teacher forcing, and its differences are gated by a deadzone calibrated separately on free-form calls, Appendix C. Probe scores are evaluated under the current policy with gradients stopped.

3.2 WAS THE LOOK NEEDED? THE DECISION VALUE

Turning to the first question, *was the look needed?* A correct answer after a call does not establish that the call was needed. The answer may already be available in h_i from the full image or the reasoning

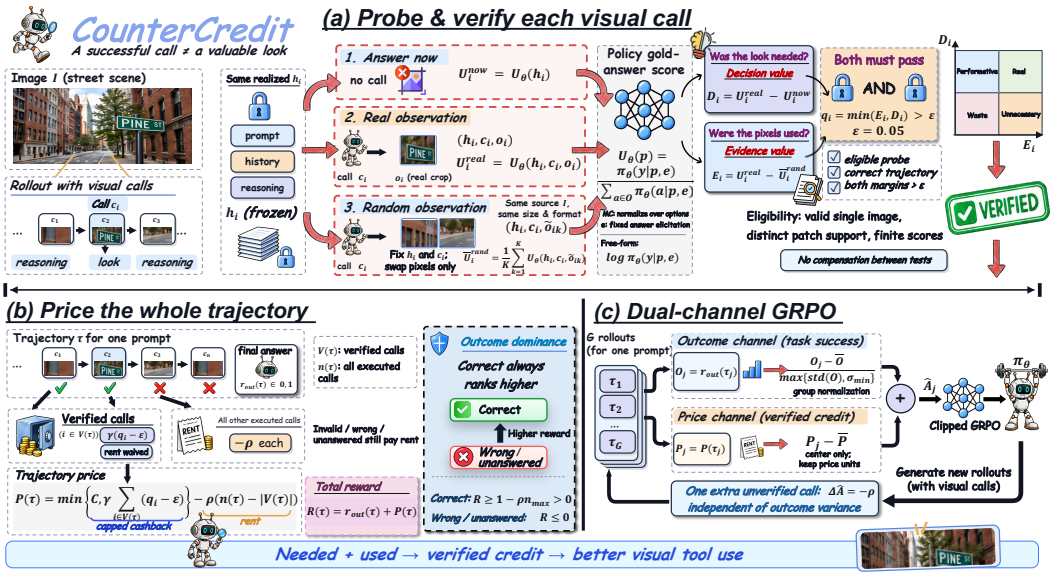


Figure 2: **CounterCredit overview**. **(a)** Each visual call is probed from its realized pre-call state with three continuations, **❶ answer now, ❷ real observation, ❸ random observation**, D_i asks whether the look was needed, E_i whether the pixels were used, and a call is verified when both exceed ε on a correct trajectory. **(b)** The price adds capped cashback on verified calls and rent on other executed calls. **(c)** Dual-channel GRPO normalizes the outcome channel and only centers the price channel.

written before the call. To answer the first question, we compare the gold-answer score of the realized visual branch with that of answering immediately from the same state. The decision value is

$$U_i^{\text{real}} = U_\theta(h_i, c_i, o_i), \qquad U_i^{\text{now}} = U_\theta(h_i), \qquad D_i := U_i^{\text{real}} - U_i^{\text{now}}. \tag{2}$$

A positive D_i means that the realized visual branch increased the gold-answer score relative to answering from h_i. Any additive score contribution that depends only on the shared pre-call prefix cancels in D_i, as formalized in Lemma 1 of Appendix A.

3.3 Was the look used? The evidence value

Turning to the second question, *were the returned pixels used?* A useful branch does not imply that its returned pixels caused the improvement. The call itself, or simply receiving an observation, may affect the gold-answer score even when the visual content is uninformative. To answer the second question, we hold the prefix h_i and the call c_i, including its text arguments, fixed and replace only the returned pixels with random patches. The evidence value is

$$\bar{U}_i^{\text{rand}} = \frac{1}{K}\sum_{k=1}^{K} U_\theta(h_i, c_i, \tilde{o}_{ik}), \qquad E_i := U_i^{\text{real}} - \bar{U}_i^{\text{rand}}. \tag{3}$$

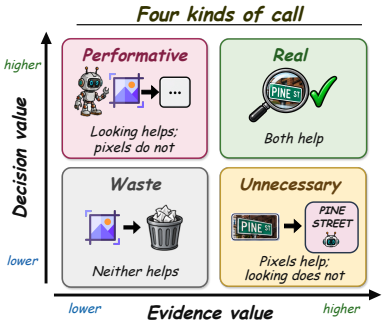


Figure 3: **The four call regions** by evidence value E_i and decision value D_i for each visual call.

E_i measures the gain in gold-answer score from the returned pixels relative to the mean score of the random-observation continuations. D_i compares the real crop with answering now, and E_i compares it with random patches of the same image, so $D_i > 0$ says the look improved the answer and $E_i > 0$ says the improvement came from the returned pixels. Needed and used denote score gains beyond the deadzone over answering now and over random patches, including added confidence in answers the policy would already get right. Both conditions must hold, as formalized in Section 3.4. Table 13 in Appendix F.5 reports used-label agreement with the default across reference sample counts, patch locations, and reference types.

3.4 BOTH MUST PASS: VERIFICATION AND PRICING

The pair (E_i, D_i) induces the credit geometry of Figure 3. Calls with $E_i > 0$ and $D_i > 0$ are *real*: looking improved the answer and the pixels mattered. Calls with $D_i > 0$ but $E_i \le 0$ are *performative*: random patches would have helped as much. Calls with $E_i > 0$ but $D_i \le 0$ are *unnecessary*: the pixels are informative, but the policy could already answer. Calls with neither effect are *waste*. Both failure regions are populated on the cold-start checkpoint, Figure 1(b), and CounterCredit pays only the real region, with credit limited by both references. Proofs of the propositions below are in Appendix A, together with the assumptions and scope of each result.

Paying a call requires one number that summarizes both effects, so we define its credit as

$$q_i = \min\{E_i, D_i\}. \tag{4}$$

The following proposition justifies the minimum.

Proposition 1 (Robust advantage over two references)**.** *For $\lambda \in [0,1]$ let $B(\lambda) = \lambda U_i^{\mathrm{now}} + (1-\lambda)\bar{U}_i^{\mathrm{rand}}$, the expected score of a reference that answers at once with probability λ and otherwise makes the call with a random patch. Then $\inf_\lambda[U_i^{\mathrm{real}} - B(\lambda)] = U_i^{\mathrm{real}} - \max\{U_i^{\mathrm{now}}, \bar{U}_i^{\mathrm{rand}}\} = q_i$, and $q_i = \sup\{m : U_i^{\mathrm{real}} \ge U_i^{\mathrm{now}} + m,\ U_i^{\mathrm{real}} \ge \bar{U}_i^{\mathrm{rand}} + m\}$.*

The minimum is therefore the advantage that survives the least favourable weighting of the two references. Verification requires this advantage to exceed a deadzone ε, fixed before RL by a null calibration on contrasts built from random patches alone and held fixed during training; Appendix C gives the procedure, the false-verification rate at each grid value, and the separate values used for multiple-choice and free-form calls. The *spurious* calls $\{i : q_i \le \varepsilon\}$ comprise the performative, unnecessary, and waste regions of Figure 3 together with the band of the real region within ε of either axis. Let $a_i \in \{0,1\}$ indicate probe eligibility, requiring a valid single-image return, distinct random-patch support, and finite probe scores. Call i is verified when

$$v_i = r_{\mathrm{out}}(\tau)\, a_i\, \mathbb{I}[q_i > \varepsilon]. \tag{5}$$

Verification alone does not discourage indiscriminate tool use, so every unverified executed call pays rent ρ, including failed parses and calls on wrong or unanswered trajectories. Verified calls are exempt and receive cashback proportional to their margin above the deadzone, capped at C per trajectory, Figure 2(b). With $\mathcal{V}(\tau) = \{i : v_i = 1\}$,

$$P(\tau) = \min\Big\{C,\ \gamma \sum_{i \in \mathcal{V}(\tau)} (q_i - \varepsilon)\Big\} - \rho\big(n(\tau) - |\mathcal{V}(\tau)|\big), \qquad R(\tau) = r_{\mathrm{out}}(\tau) + P(\tau). \tag{6}$$

For an eligible call on a correct trajectory, crossing ε removes the rent, so payment jumps by ρ at the deadzone and then grows with q_i until the cap binds. The price is added to the outcome reward, so it could in principle rank a cheap wrong trajectory above an expensive correct one. The following proposition rules this out.

Proposition 2 (Outcome dominance)**.** *Suppose every trajectory executes at most $n_{\max}$ calls and $\rho\, n_{\max} < 1$. A correct trajectory has $R(\tau) \ge 1 - \rho\, n_{\max}$, whereas a wrong or unanswered trajectory has $R(\tau) = -\rho\, n(\tau) \le 0$, so every correct trajectory outranks any wrong or unanswered one by at least $1 - \rho\, n_{\max}$, regardless of their respective call counts.*

The bound holds because cashback is nonnegative and restricted to correct trajectories while total rent is at most $\rho\, n_{\max}$; with $\rho = 0.05$ and $n_{\max} = 6$ the margin is 0.7, and $C = 0.7$ keeps correct-trajectory rewards in $[0.7, 1.7]$; the guarantee weakens linearly in the call budget and vanishes at $\rho\, n_{\max} = 1$. Rent also covers verification noise. With $K = 3$ the random-patch mean fluctuates, so q_i is a noisy estimate of the credit $q_i^\star$ that the same call would receive against the expected patch score, and a crop with $q_i^\star \le \varepsilon$ can still clear the deadzone by chance. The following proposition shows that the rent makes such luck unprofitable.

Proposition 3 (Chance verification does not pay)**.** *For an executed call let $A_i = \{q_i^\star \le \varepsilon\}$ be the event that its value at the reference mean falls short of the verification margin, and suppose $\Pr(A_i) > 0$ and $\Pr(v_i = 1 \mid A_i) \le \alpha \in [0,1)$. Its uncapped price satisfies $\mathbb{E}[X_i \mid A_i] \le \alpha\gamma(1-\varepsilon) - (1-\alpha)\rho$, which is negative whenever the rent satisfies $\rho > \frac{\alpha}{1-\alpha}\gamma(1-\varepsilon)$.*

The rent thus offsets in expectation what chance verification collects, at our constants for every α below 9.5%; for free-form calls the bound assumes a score range of at most B, with $\gamma(B - \varepsilon)$ in

place of $\gamma(1-\varepsilon)$. The null calibration of Appendix C, run before training on training-pool questions and reproduced on V* and HR-Bench-4K, gives proxy rates of 2.4% and 2.5% for it; Appendix A gives the proof and shows that the cap only lowers the trajectory price.

3.5 DUAL-CHANNEL GRPO OPTIMIZATION

Standard GRPO (Shao et al., 2024) normalizes total reward within each prompt group $g(j)$, which makes the effective scale of a fixed price depend on group composition: outcome variation suppresses price differences and identical outcomes amplify them. We therefore separate rollout j's reward into outcome $O_j = r_{\text{out}}(\tau_j)$ and price $P_j = P(\tau_j)$, and use

$$\widehat{A}_j = \frac{O_j - \overline{O}_{g(j)}}{\max\{\text{std}(O_{g(j)}), \sigma_{\min}\}} + \big(P_j - \overline{P}_{g(j)}\big), \tag{7}$$

with $\sigma_{\min} = 0.15$. The outcome channel retains GRPO's group-relative normalization, whereas the price channel is only centered and therefore remains in its original units, Figure 2(c).

Proposition 4 (Price scale under normalization)**.** *In a group whose rollouts are all correct, carry no verified call, and differ in their unverified call counts n_j, single-channel normalization gives $A_j = -(n_j - \bar{n})/\text{std}(n)$, independent of ρ, whereas Equation 7 gives $\widehat{A}_j = -\rho\,(n_j - \bar{n})$.*

Standard GRPO thus cannot see the rent's magnitude in such groups, while the price channel keeps it: an otherwise identical rollout with one additional unverified call differs in advantage by exactly ρ. When no rollout is correct the price advantage is set to zero, since prices there carry only call counts. In a group holding both outcomes, Proposition 2 carries over: a correct rollout's $\widehat{A}_j$ exceeds a wrong one's by at least $1/\max\{\text{std}(O), \sigma_{\min}\} - \rho\, n_{\max}$, which is at least 1.5 since the sample standard deviation of eight binary outcomes is at most 0.54.

We substitute $\widehat{A}_j$ into the clipped GRPO objective without backpropagating through probe scores; a correct trajectory with m eligible calls costs $m(K+2)$ score evaluations, Algorithm 1 in Appendix C.

4 EXPERIMENTS

4.1 SETUP

Training. Qwen2.5-VL-7B-Instruct (Bai et al., 2025) is fine-tuned on 6,960 filtered Mini-o3 trajectories (Lai et al., 2026), then trained with GRPO under the reward of Section 3 on 16,000 VisualProbe and DeepEyes prompts (Lai et al., 2026; Zheng et al., 2026), deduplicated, decontaminated against every evaluation image, and filtered by difficulty. The outcome reward r_{out} counts an answer correct when it matches the reference exactly or by option text; free-form answers that fail both are judged by GPT-4o (OpenAI, 2024), whose judgments agree with human labels on 99.4% of a 500-answer sample, Appendix D. Appendix B details curation and Appendix C the configuration.

Benchmarks and metrics. Evaluation covers four perception benchmarks, V* (Wu & Xie, 2024), HR-Bench-4K and HR-Bench-8K (Wang et al., 2025b), and MME-RealWorld (Zhang et al., 2025b), and three general benchmarks, MMStar (Chen et al., 2024), ChartQA (Masry et al., 2022), and BLINK (Fu et al., 2024). On V* and both HR-Bench splits, the *spurious-call rate* is the share of executed image-returning calls with $\min\{E_i, D_i\} \le \varepsilon$, with each agent's E_i and D_i computed under its own gold-answer score on its own prefixes. Appendix D gives the evaluation details.

Baselines. Baselines are seven open tool-using agents re-evaluated from their released checkpoints in the same pipeline, DeepEyes, Pixel-Reasoner, Mini-o3, Thyme, CodeV-RL, DeepEyesV2 (Hong et al., 2026), and PyVision-RL (Zhao et al., 2026a); three models without visual tools, Qwen2.5-VL-7B, Qwen3-VL-8B (Qwen Team, 2025), and InternVL3-8B (Zhu et al., 2025); and GPT-5 (OpenAI, 2025) and Gemini-2.5-Pro (Gemini Team, Google, 2025) through their APIs. Appendix D gives the audit rule for code-based agents. TACO has no public checkpoint; Figure 1(b) audits its branch probe and Table 3 retrains its branch reward.

Table 1: Accuracy (%) on four perception and three general benchmarks, all evaluated under our protocol; Avg. is the mean over the seven. Bold: best per column among tool-using agents; underline: second best. Cell colour: gain over Qwen2.5-VL-7B. The ours row is the run used throughout; ± is the sample standard deviation over three seeds, Table 12 (Appendix F.4). Baselines are single runs, and their spread is not measured.

	Perception				General			
Model	**V***	**HR-4K**	**HR-8K**	**MME-RW**	**MMStar**	**ChartQA**	**BLINK**	**Avg.**
Closed models (API)								
GPT-5	72.3	75.6	73.8	68.9	73.4	76.5	69.7	72.9
Gemini-2.5-Pro	79.1	84.2	81.1	58.7	73.4	83.8	73.6	76.3
Open models without a visual tool								
Qwen2.5-VL-7B	77.0	69.1	65.6	58.0	64.4	83.9	56.2	67.7
Qwen3-VL-8B	85.9	79.2	74.3	60.7	70.5	88.4	68.9	75.4
InternVL3-8B	70.2	70.4	69.1	61.7	68.2	86.0	55.6	68.7
Tool-using agents (7B)								
DeepEyes	84.8	73.4	70.9	64.3	65.1	82.6	57.4	71.2
Pixel-Reasoner	83.8	70.6	67.2	64.5	64.8	81.9	56.9	70.0
Mini-o3	87.4	78.0	73.4	65.2	64.1	85.3	55.6	72.7
Thyme	83.8	76.8	71.3	64.7	65.2	85.9	56.4	72.0
CodeV-RL	84.8	76.3	71.5	64.6	67.2	84.1	58.3	72.4
DeepEyesV2	83.2	77.6	73.2	64.9	65.4	86.3	57.8	72.6
PyVision-RL	86.9	77.3	73.8	62.4	65.7	86.2	57.9	72.9
CounterCredit (ours)	**89.5**±0.81	**80.2**±0.75	**76.4**±0.90	**67.7**±0.50	**68.9**±0.35	**87.1**±0.30	**62.8**±0.65	**76.1**±0.40

4.2 Main Results

The lead spans all seven benchmarks. CounterCredit leads every tool agent on all seven benchmarks in Table 1, although the agents differ in tools, prompts, and interaction formats, so the matched comparisons carry the causal claim. On V*, HR-Bench-4K, and HR-Bench-8K it adds 12.5, 11.1, and 10.8 points over the base and exceeds the best tool agent by 2.1, 2.2, and 2.6. The three general benchmarks rise as well, by 3.2 to 6.6 points, the largest gains among the tool agents, and the three resolution scores exceed GPT-5, which has no tool loop.

CounterCredit pairs its accuracy with selective tool use. Table 2 places its call count in the middle of the field, 1.76 to 1.87 per question, with the lowest spurious-call rate, 31.4% to 36.3%, each judged under its own gold-answer score; re-scoring the same V* rollouts with one common scorer, the SFT cold start, and with an outside scorer that took no part in training leaves CounterCredit lowest under both and reorders the seven baselines by at most one adjacent swap, Spearman $\rho = 0.96$ and 0.75 against their own scorers, Appendix F.

Mini-o3 is the matched comparison. It uses the same crop tool and cold-start data under an outcome reward from an LLM judge: on V* CounterCredit scores 2.1 points higher, makes 1.78 calls per question against 2.50, and lowers the spurious-call rate from 40.7% to 31.4%. Averaged over the three resolution benchmarks, Figure 4 places CounterCredit beyond the observed baseline frontier, at 82.0 points in 8.1 seconds against 74 to 78 points for the agents under six seconds and 79.6 for the best baseline, which takes 10.9.

4.3 Auditing Existing Rewards

Most calls pass neither test. Figure 1(b) audits four rewards on the SFT checkpoint's trajectories: the outcome reward, which credits every call on a correct trajectory; the branch probe of TACO (Feng et al., 2026); the external crop judge of CodeV (Hou et al., 2026), served by GPT-4o (OpenAI, 2024); and the self-judge of FaithEyes (Wang et al., 2026c), answered by the policy itself. Base rates are low: 14% to 18% of executed calls pass each test alone and 10% to 12% pass both; Appendix F gives each payment rule and the numbers per benchmark.

The two tests are not one test asked twice. Only 57% to 85% of needed calls were also used and 54% to 78% of used calls were needed, so a reward that checks one test alone pays 15% to 46% of its verified calls for the wrong reason. Re-decoding the same calls with a random patch in place of the real crop separates them at the answer level too: real calls change the decoded answer sixteen times

Table 2: Calls per question, latency in seconds per question at batch size 1 on one NVIDIA H20 with vLLM, and spurious-call rate, the share of executed image-returning calls with $\min(E, D) \leq \varepsilon$. Bold: lowest; underline: second lowest.

	Calls / question			Latency (s)			Spurious-call rate (%)		
Model	**V***	**HR-4K**	**HR-8K**	**V***	**HR-4K**	**HR-8K**	**V***	**HR-4K**	**HR-8K**
DeepEyes	0.91	0.98	0.96	16.1	14.9	15.8	87.4	83.6	82.1
Pixel-Reasoner	0.95	0.84	0.82	2.0	2.0	3.0	68.1	52.5	52.1
Mini-o3	2.50	2.46	2.60	7.3	12.1	13.4	40.7	<u>50.5</u>	<u>51.0</u>
Thyme	0.14	0.19	0.22	2.3	2.6	2.9	<u>36.1</u>	77.2	75.3
CodeV-RL	1.84	0.43	0.47	6.7	3.3	3.6	52.4	73.7	53.9
DeepEyesV2	1.23	1.27	1.32	4.4	5.7	6.3	51.3	55.3	55.3
PyVision-RL	2.87	3.30	3.60	14.3	20.0	23.0	54.6	56.4	63.9
CounterCredit (ours)	1.78	1.76	1.87	5.4	8.8	10.0	**31.4**	**35.1**	**36.3**

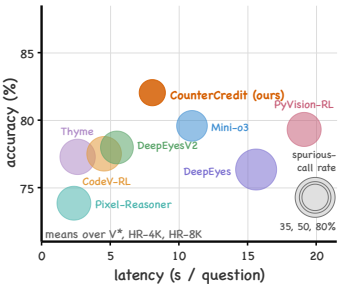


Figure 4: Accuracy against latency; marker area is the spurious-call rate.

Table 3: **Training stages and reward ablation.** All RL rows start from the SFT checkpoint with the same prompt pool and budget and differ only in the reward; $\gamma = 0$ removes the cashback, $\rho = 0$ the rent, and the other arms are defined in the text. Avg.: mean of the seven benchmarks; calls and spurious-call rate on V*. The CounterCredit row is the seed shared with every arm; its spread over three seeds is in Tables 1 and 12.

	Perception				General				V* calls	
Model	**V***	**HR-4K**	**HR-8K**	**MME-RW**	**MMStar**	**ChartQA**	**BLINK**	**Avg.**	**Calls**	**Spur. %**
Starting points										
Qwen2.5-VL-7B	77.0	69.1	65.6	58.0	64.4	83.9	56.2	67.7	0	–
SFT (cold start)	71.7	70.4	61.6	55.6	60.3	80.8	54.5	65.0	2.68	88.2
Which calls are verified										
Outcome-only GRPO	80.1	73.9	69.4	56.8	61.7	86.4	54.3	68.9	1.84	82.1
Branch reward (TACO)	85.1	76.2	72.0	60.7	62.8	86.9	57.4	71.6	2.45	62.9
Verify on E only	82.2	74.6	68.3	59.0	62.3	86.4	53.1	69.4	1.46	71.3
Verify on D only	84.3	76.5	72.5	60.5	63.5	86.5	55.5	71.3	1.65	65.0
What verification buys										
CounterCredit, $\gamma = 0$	81.2	74.6	70.2	59.4	62.9	86.3	55.4	70.0	1.35	62.2
CounterCredit, $\rho = 0$	85.9	77.9	74.9	63.0	65.3	86.9	58.0	73.1	2.70	48.7
How the price enters training										
Single-channel GRPO	62.8	68.8	63.9	57.1	60.0	83.7	55.3	64.5	0.78	62.5
CounterCredit	89.5	80.2	76.4	67.7	68.9	87.1	62.8	76.1	1.78	31.4

as often as waste, and performative calls change it as rarely as waste although their decision value is positive, Table 8 (Appendix F.1).

None of the four rewards asks both questions. On V* the outcome reward's payments pass both tests 12.8% of the time against a base rate of 11.8%, and the two judges, which read the crop, reach 21.0% and 14.7%, with HR-Bench in Table 7 (Appendix F.1). None of the three changes the call it scores, and 56% to 77% of their payments pass neither test. The branch probe of TACO does change it, and one intervention buys precision: 42% to 50% of its payments pass both tests, at 5 to 7 per hundred. It removes the whole branch, so it cannot see whether the pixels mattered: 19.4% of its payments on V* go to performative calls, the region the second question closes.

4.4 Ablation

Accuracy depends on which calls survive, not on how many. The SFT checkpoint calls 2.68 times per question on V*, 88.2% of those calls are spurious, and its average falls 2.7 points below the base, Table 3. Outcome-only GRPO cuts the calls by a third and gains 1.2 points over the base, yet reduces the spurious share only to 82.1%: a reward that pays every call on a correct trajectory equally changes how often the policy calls, not which calls survive. CounterCredit averages 76.1 against 68.9 at 1.78 calls per question against 1.84, and the spurious share falls to 31.4%. Every RL row shares the checkpoint, prompt pool, budget, decoding, and seed, so this comparison carries the paper's causal claim; outcome-only GRPO uses our cold start and six-call budget, whereas the released Mini-o3 of Table 1 uses its own.

No component of the reward is redundant. Verifying on E_i or on D_i alone recovers 0.5 and 2.4 of the 7.2-point gain and leaves 71.3% and 65.0% of calls spurious, since each single test leaves one

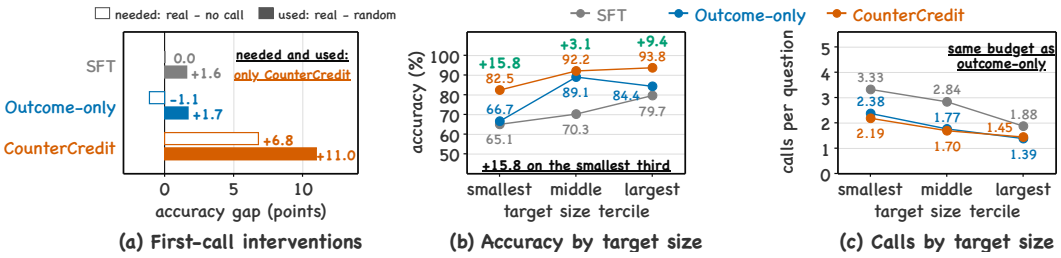


Figure 5: V* by first-call intervention and by target size for the SFT checkpoint, outcome-only GRPO, and CounterCredit. (a) Trajectories stopped at the first call and answered from the cached prefix, with the two gaps defined in the legend. (b) Accuracy and (c) calls per question over terciles of the target's share of the image; green labels give CounterCredit's margin over outcome-only GRPO; values in Tables 9 and 10 (Appendix F.2).

failure region unpriced. A branch reward in the style of TACO (Feng et al., 2026), which credits a call when the tool branch turns a wrong answer right, recovers 2.7 at 62.9% spurious and needs 2.45 calls per question: asking the first question alone leaves 4.5 to 4.8 points on the table however it is asked. Removing the cashback recovers 1.1 points at 1.35 calls per question and removing the rent recovers 4.2 at 2.70: the rent governs how often the policy calls and the cashback governs which calls it makes. Normalizing the whole reward instead of Equation 7 is worse still, reducing the policy to 0.78 calls and 64.5 on average, below the cold start. Proposition 4 gives the scale that normalization erases in unanimous groups, and the single-channel arm also lacks the floor $\sigma_{\min}$, so more than the normalization differs between the two.

The arms separate only after the returned pixels arrive. Figure 5 runs two checks that do not apply the verification rule the reward is built from, so the gains they show do not depend on the metric the policy was trained against. In (a) the three arms lie within 3.1 points of each other before the pixels arrive. The real crop is then worth 0.0 and -1.1 points relative to answering at once for the two baselines and 1.6 and 1.7 relative to a random patch, while for CounterCredit the same margins are 6.8 and 11.0. In (b) the gain concentrates on the smallest third of targets, 15.8 points over outcome-only GRPO against 3.1 and 9.4 on the middle and largest thirds, where the evidence is a smaller part of the image and a look pays only when it fetches that evidence.

The recipe transfers to a second base. Table 4 applies the same cold start and reward without change: the average rises from 67.7 to 76.1 on Qwen2.5-VL-7B and from 75.4 to 80.8 on Qwen3-VL-8B, with the largest gains where the base is weakest; no reward ablation was run on Qwen3-VL-8B.

Table 4: **Two base models on the perception benchmarks (accuracy %).** Base rows answer directly; + rows apply our cold start and RL. Avg. is over all seven benchmarks; the three general benchmarks are in Table 11 (Appendix F.3).

Base / Method	V*	HR-4K	HR-8K	MME-RW	Avg.
Qwen2.5-VL-7B	77.0	69.1	65.6	58.0	67.7
+ CounterCredit	89.5	80.2	76.4	67.7	76.1
Qwen3-VL-8B	85.9	79.2	74.3	60.7	75.4
+ CounterCredit	91.6	85.4	84.0	70.1	80.8

5 Conclusion and Limitations

CounterCredit pays a visual call only when looking helped and the returned pixels carried the help, and charges rent otherwise. On the cold start only 9.8% to 11.8% of calls met both conditions. Paying for those alone raised accuracy on the three resolution benchmarks by 6.3 to 9.4 points over outcome-only GRPO from the same cold start and lifted a second base model; against a matched branch reward it reaches 76.1 on the seven-benchmark average against 71.6, at 1.78 calls per question against 2.45 and 31.4% spurious calls against 62.9%. The decision value does not separate the information in the crop from the effect of eliciting an answer at a prefix where the policy had decided to crop. Two further limits set the next steps. The evidence value is defined only for a crop, which admits a matched null return; tools that return text or structured output need their own null before the same question can be asked of them. The efficiency metric is built from the two contrasts the reward is trained on, so an evaluation defined apart from the reward would settle the question.

Reproducibility Statement

Section 4.1 gives the data sources, the base models, the judge, and the evaluation pipeline; Appendix B the curation and pool-filtering rules; Appendix C the reward constants, the deadzone calibration, the full cold-start and RL configuration, and Algorithm 1; Appendix D the per-benchmark scoring rules; Appendix G further limitations. Every benchmark is public and every baseline was re-evaluated from its released checkpoint under the same pipeline. We will release the code, configurations, calibration and audit scripts, and the probe logs behind the tables.

Ethics Statement

This work trains and evaluates models on publicly released datasets and benchmarks. It involves no human subjects, no personal data, and no new data collection or annotation. The reward is computed from the policy's own answer distribution and introduces no external judge at training time beyond the correctness rule of Section 4.1.

The Use of Large Language Models

Large language models were used to polish the writing and to assist with plotting and formatting scripts. The problem formulation, the method, the experiments, and the claims are the authors' own.

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.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024. arXiv:2403.20330.

Yongchao Chen, Jiefeng Chen, Rui Meng, Ji Yin, Na Li, Chuchu Fan, Chi Wang, Tomas Pfister, and Jinsung Yoon. TUMIX: Multi-agent test-time scaling with tool-use mixture. In *International Conference on Learning Representations*, volume 2026, pp. 55848–55878, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5b755cf5598a4324d253025e1fbbba52-Paper-Conference.pdf.

Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, et al. Sensesearch: Empowering vision-language models with high-resolution agentic search-reasoning via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26508–26517, 2026.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *ACM Multimedia*, 2024. arXiv:2407.11691.

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*, 2025.

Mingkuan Feng, Jinyang Wu, Hao Gu, Fangrui Lv, Ruihan Jin, Chuyuan Zhang, Zhengqi Wen, and Jianhua Tao. TACO: Tool-augmented credit optimization for agentic tool use. *arXiv preprint arXiv:2606.30251*, 2026.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *ECCV*, 2024. arXiv:2404.12390.

Gemini Team, Google. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Jiawei Guo, Donglei Yu, Yu Chen, Xiang Wang, Shuai Li, Xinpei Zhao, Huaxing Liu, Qinghao Wang, and Minpeng Liao. Do multimodal agents really benefit from tool use? a systematic study of capability gains, 2026.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. In *International Conference on Learning Representations*, 2026. arXiv:2511.05271.

Xinhai Hou, Shaoyuan Xu, Manan Biyani, Moyan Li, Jia Liu, Todd C Hollon, and Bryan Wang. Codev: Code with images for faithful visual reasoning via tool-aware policy optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21500–21510, June 2026.

Minki Kang, Shizhe Diao, Ryo Hachiuma, Sung Ju Hwang, Pavlo Molchanov, Yu-Chiang Frank Wang, and Byung-Kwan Lee. Agent explorative policy optimization for multimodal agentic reasoning, 2026.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 76722–76746, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7c40c5050bd029a3ea7ff8b01412f735-Paper-Conference.pdf.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Xixuan Song, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. Visualagentbench: Towards large multimodal models as visual foundation agents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 95650–95707, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/eea71dc576381b88f2a0ca4dedc2140d-Paper-Conference.pdf.

Zujing Liu, Junwen Pan, Qi She, Yuan Gao, and Guisong Xia. On the faithfulness of visual thinking: Measurement and enhancement. *arXiv preprint arXiv:2510.23482*, 2025b.

Yan Ma, Weiyu Zhang, Tianle Li, Linge Du, Xuyang Shen, and Pengfei Liu. What does vision tool-use reinforcement learning really learn? disentangling tool-induced and intrinsic effects for crop-and-zoom. In *International Conference on Machine Learning*, 2026. arXiv:2602.01334.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

OpenAI. GPT-5 system card. https://openai.com/index/gpt-5-system-card/, 2025.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Accessed: 2026-09-18.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

Qwen Team. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Vishwas Sathish, Viresh Ranjan, Xinliang Zhu, Arnab Dhua, and Douglas Gray. Eliciting self-verification in multimodal reasoning agents with reinforcement learning. In *European Conference on Computer Vision*, 2026. arXiv:2609.08025.

Jiahao Shao, Yuanbo Yang, Yiyi Liao, Yujun Shen, Ceyuan Yang, and Yinghao Xu. Thinking with tools, not with pixels: Tool calls as text scaffolds for visual reasoning, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 8222–8251, 2025. doi: 10.52202/085713-0280. URL https://openreview.net/forum?id=VeZkY3JjWV.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *NeurIPS*, 2023.

Changpeng Wang, Haozhe Wang, Xi Chen, Junhan Liu, Taofeng Xue, Chong Peng, Donglian Qi, Fangzhen Lin, and Yunfeng Yan. From illusion to intention: Visual rationale learning for vision-language reasoning. *arXiv preprint arXiv:2511.23031*, 2025a.

Chaoyang Wang, Kaituo Feng, Dongyang Chen, Zhongyu Wang, Zhixun Li, Sicheng Gao, Meng Meng, Xu Zhou, Manyuan Zhang, Yuzhang Shang, and Xiangyu Yue. AdaTooler-V: Adaptive tool-use for images and videos. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18058–18072, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.898. URL https://aclanthology.org/2026.findings-acl.898/.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *International Conference on Learning Representations*, 2026b. arXiv:2510.14967.

Haoqing Wang, Xingrun Xing, Wei Xia, Ziheng Li, and Yehui Tang. FaithEyes: Towards faithful tool use via multi-agent process-image verification. *arXiv preprint arXiv:2607.28225*, 2026c.

Haozhe Wang, Qixin Xu, Changpeng Wang, Taofeng Xue, Chong Peng, Wenhu Chen, and Fangzhen Lin. Bad seeing or bad thinking? rewarding perception for multimodal reasoning. In *Forty-third International Conference on Machine Learning*, 2026d. URL https://openreview.net/forum?id=duzdftKtUA.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *AAAI*, 2025b. arXiv:2408.15556.

Zhiheng Wang, Bo Peng, Lai Wei, and Chaochao Lu. The illusion of visual tool-use: A causal audit of thinking with images. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026e. arXiv:2608.06270.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 78298–78319, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7e6f445a74cdb71931aac64f1e3f49c9-Paper-Conference.pdf.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *CVPR*, 2024. arXiv:2312.14135.

Xi Xiao, Chen Liu, Chih-Ting Liao, Yunbei Zhang, Qizhen Lan, Yuxiang Wei, Lin Zhao, Janet Wang, Jianyang Gu, Muchao Ye, Tianyang Wang, and Hao Xu. Staying VIGILant: Mitigating visual laziness via counterfactual visual alignment in MLLMs. In *European Conference on Computer Vision*, 2026. arXiv:2606.26387.

Eunseop Yoon, Hee Suk Yoon, Jaehyun Jang, SooHwan Eom, Qi Dai, Chong Luo, Mark A. Hasegawa-Johnson, and Chang D. Yoo. PACR: Progressively ascending confidence reward for LLM reasoning. *arXiv preprint arXiv:2510.22255*, 2025.

Genpei Zhang. The visual insensitivity gap: Diagnosing when vision-language models fail to use visual evidence, 2026.

Miaosen Zhang, Qi Dai, Yifan Yang, Jianmin Bao, Dongdong Chen, Kai Qiu, Chong Luo, Xin Geng, and Baining Guo. MageBench: Bridging large multimodal models to agents. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1415–1427, 2026a. arXiv:2412.04531.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient VLMs, 2025a.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, 2025b. arXiv:2408.13257.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images. In *International Conference on Learning Representations*, 2026b. arXiv:2508.11630.

Shitian Zhao, Shaoheng Lin, Ming Li, Haoquan Zhang, Wenshuo Peng, Kaipeng Zhang, and Chen Wei. PyVision-RL: Forging open agentic vision models via RL. *arXiv preprint arXiv:2602.20739*, 2026a.

Xuanpu Zhao, Zhentao Tan, Dianmo Sheng, Tianxiang Chen, Yao Liu, Yue Wu, Tao Gong, Qi Chu, and Nenghai Yu. Learning to focus and precise cropping:a reinforcement learning framework with information gaps and grounding loss for mllms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25938–25947, June 2026b.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/cdb347d7516d52a9280ea9d6708911f6-Paper-Conference.pdf.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Mingyu Liu, Zheng Huang, Anzhou Li, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, and Chunhua Shen. ACTIVE-o3 : Empowering MLLMs with active perception via pure reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=1uU7X9zzJB.

APPENDIX

A THEORETICAL SUPPORT

This appendix proves the lemma and the four propositions of Section 3 and states what each one buys and where it stops. The results are exact algebra on the reward as implemented, together with one expectation bound whose premise is measured rather than enforced.

Summary

Robustness. q_i is the advantage a crop keeps against the worst mixture of the two references (Proposition 1).
Safety. The price never reorders a correct and a wrong trajectory; the margin is 0.7 (Proposition 2).
Incentive. Chance verification loses in expectation whenever its rate is under 9.5% (Proposition 3).
Scale. Single-channel normalization erases the rent's size in unanimous groups; the dual channel keeps it (Proposition 4).

A.1 SETUP AND ASSUMPTIONS

Notation follows Section 3. For call i, $U_{ik}^{\mathrm{rand}} = U_\theta(h_i, c_i, \tilde{o}_{ik})$ and $\bar{U}_i^{\mathrm{rand}}$ is their mean over $k = 1, \dots, K$, so that

$$q_i = \min\{E_i, D_i\} = U_i^{\mathrm{real}} - \max\{U_i^{\mathrm{now}}, \bar{U}_i^{\mathrm{rand}}\}. \tag{8}$$

Write $b_i = \gamma\, v_i\, (q_i - \varepsilon) \geq 0$ for the pre-cap cashback of call i, so that Equation 6 reads $P(\tau) = \min\{C, \sum_i b_i\} - \rho\,(n(\tau) - |\mathcal{V}(\tau)|)$ with the cap acting on the sum. For a group of $G = 8$ rollouts, $\bar{z}$ denotes the group mean of a quantity z and $\mathrm{std}(z)$ its sample standard deviation, which divides by $G - 1$. The proofs use four assumptions.

- (A1) **Bounded scores.** $U_\theta(p) \in [0, 1]$ for every prefix p, as for the renormalized option probability of Equation 1; hence D_i, E_i, and q_i lie in $[-1, 1]$.
- (A2) **Call budget.** Every trajectory executes at most $n_{\max}$ calls, and $\rho\, n_{\max} < 1$.
- (A3) **False-verification rate.** On the event $A_i = \{q_i^\star \leq \varepsilon\}$ of Proposition 3, $\Pr(v_i = 1 \mid A_i) \leq \alpha$ with $\alpha < \rho/(\rho + \gamma(1 - \varepsilon))$, equivalently $\rho > \frac{\alpha}{1-\alpha}\gamma(1 - \varepsilon)$.
- (A4) **Group normalization.** The single-channel arm of Table 3 divides $R_j - \bar{R}$ by $\mathrm{std}(R)$ with no floor; the dual channel is Equation 7, whose outcome channel carries the floor $\sigma_{\min}$ and whose price channel is centered but not normalized.

A.2 COMMON SHIFTS CANCEL

Both values are differences of scores on the same prefix. Whatever the scorer adds to every compared arm alike, a prompt-level preference for one option letter or a call text that raises the confidence of every continuation, drops out of the difference. This is what makes the reward insensitive to text that raises every arm's score alike; TACO states the same invariance for its two-probe difference (Feng et al., 2026).

Lemma 1 (Common additive shifts cancel)**.** *Fix call i with its realized crop and sampled patches, and let primes denote quantities recomputed after a text edit. (a) If $U_i^{\mathrm{real}} = T_i^{\mathrm{real}} + s(h_i)$ and $U_i^{\mathrm{now}} = T_i^{\mathrm{now}} + s(h_i)$ for a component s that depends only on the prefix, then $D_i = T_i^{\mathrm{real}} - T_i^{\mathrm{now}}$; in particular, an edit of h_i that shifts both scores by the same δ leaves $D_i' = D_i$. (b) If h_i is fixed and an edit of c_i shifts U_i^{real} and every U_{ik}^{rand} by the same η, then $E_i' = E_i$ and $D_i' = D_i + \eta$. (c) For every call, $0 \leq b_i \leq \gamma \max\{E_i - \varepsilon, 0\}$; under the edit of (b) the same bound holds for b_i', and with the other calls' contributions fixed, including call i raises the capped sum by an amount in $[0, b_i]$.*

Proof. (a) By definition, $D_i = (T_i^{\mathrm{real}} + s(h_i)) - (T_i^{\mathrm{now}} + s(h_i)) = T_i^{\mathrm{real}} - T_i^{\mathrm{now}}$. An equal shift of both scores is the case $s(h_i') = s(h_i) + \delta$ with T_i^{real} and T_i^{now} unchanged, so $D_i' = D_i$. (b) Averaging preserves the shift, $(\bar{U}_i^{\mathrm{rand}})' = \frac{1}{K}\sum_k (U_{ik}^{\mathrm{rand}} + \eta) = \bar{U}_i^{\mathrm{rand}} + \eta$, so $E_i' = (U_i^{\mathrm{real}} + \eta) - (\bar{U}_i^{\mathrm{rand}} + \eta) = E_i$. The answer-now score $U_i^{\mathrm{now}} = U_\theta(h_i)$ does not contain c_i and is untouched, so $D_i' = D_i + \eta$. (c) If $v_i = 0$ then $b_i = 0$. If $v_i = 1$ then $q_i > \varepsilon$ and $E_i \geq q_i$ by Equation 8, so

$0 < b_i = \gamma(q_i - \varepsilon) \le \gamma(E_i - \varepsilon) = \gamma \max\{E_i - \varepsilon, 0\}$. Under (b), $q_i' = \min\{E_i, D_i + \eta\} \le E_i$, and the same two cases bound b_i'. Let $S \ge 0$ be the pre-cap cashback of the other calls; the capped sum with and without call i differs by $\min\{C, S + b_i\} - \min\{C, S\} = \min\{b_i, \max\{C - S, 0\}\} \in [0, b_i]$, which completes the proof. □

A reward that paid the absolute score U_i^{real}, or the sum $U_i^{\text{real}} + \bar{U}_i^{\text{rand}}$, would move by δ or 2δ under the same shift and could be farmed by call text alone; under CounterCredit the shift cancels in the value it enters. A call-text shift can still raise D_i and with it the cashback, but never above $\gamma \max\{E_i - \varepsilon, 0\}$, which that shift leaves unchanged; this is the text-laundering row of Table 6. The lemma does not make D_i invariant to every edit of the prefix: reasoning written before the call that helps only the real branch is evidence, and D_i should move with it. What the hypothesis excludes is movement common to the compared arms; any other score change reaches the cashback only through the verification gates, the deadzone, and the cap.

A.3 The minimum is a worst-case advantage

The two references answer different questions, and a reward could combine them in several ways: pay E_i, pay D_i, or pay an average. Proposition 1 singles out the minimum as the combination that a skeptic holding either reference cannot dispute. It is the advantage the crop keeps against every mixture of the two references, and the largest margin by which the real crop beats both at once.

Proposition 1 *(restated). For $\lambda \in [0, 1]$ let $B(\lambda) = \lambda U_i^{\text{now}} + (1 - \lambda)\bar{U}_i^{\text{rand}}$. Then $\inf_\lambda [U_i^{\text{real}} - B(\lambda)] = q_i$, and $q_i = \sup\{m : U_i^{\text{real}} \ge U_i^{\text{now}} + m,\ U_i^{\text{real}} \ge \bar{U}_i^{\text{rand}} + m\}$.*

Proof. Write $u = U_i^{\text{real}}$, $w = U_i^{\text{now}}$, and $r = \bar{U}_i^{\text{rand}}$. For $\lambda \in [0, 1]$,

$$u - B(\lambda) = u - \lambda w - (1 - \lambda) r = \lambda(u - w) + (1 - \lambda)(u - r) = \lambda D_i + (1 - \lambda) E_i,$$

which is affine in λ. An affine function on $[0, 1]$ attains its infimum at an endpoint: at $\lambda = 1$ with value D_i when $w \ge r$, and at $\lambda = 0$ with value E_i when $w < r$. Hence $\inf_\lambda [u - B(\lambda)] = \min\{D_i, E_i\} = u - \max\{w, r\} = q_i$, the form of Equation 8. For the second claim, $u \ge w + m$ and $u \ge r + m$ hold exactly when $m \le u - w$ and $m \le u - r$, so the feasible set is $(-\infty, \min\{D_i, E_i\}] = (-\infty, q_i]$ and its supremum is q_i, which completes the proof. □

Taking the minimum is therefore not a tie-break between two tests. A fixed average $\lambda D_i + (1 - \lambda) E_i$ would pay a call that helps against one reference and hurts against the other: a performative call with $D_i = 0.4$ and $E_i = -0.1$ has value 0.15 under $\lambda = \frac{1}{2}$, which would earn cashback 0.05 if the other gates pass, and value -0.1 under the minimum, which earns nothing and pays rent. The endpoint argument uses only that the objective is affine in the mixing weight, so a third reference, such as the gray patch or the blurred crop of Table 13, would enter through the maximum in Equation 8 in the same way.

A.4 The price never overrides the outcome

The price is added to a binary outcome, so the first thing to check is that it cannot flip the ranking the outcome induces. Proposition 2 bounds the price from both sides and shows that the bounds leave a gap.

Proposition 2 *(restated). Under (A2), a correct trajectory has $R(\tau) \ge 1 - \rho\, n_{\max}$, a wrong or unanswered trajectory has $R(\tau) = -\rho\, n(\tau) \le 0$, and every correct trajectory outranks any wrong or unanswered one by at least $1 - \rho\, n_{\max}$.*

Proof. Every verified call has $q_i > \varepsilon$, so every $b_i \ge 0$ and $0 \le \min\{C, \sum_i b_i\} \le C$. By (A2), $0 \le \rho\,(n(\tau) - |\mathcal{V}(\tau)|) \le \rho\, n(\tau) \le \rho\, n_{\max}$. Together, $-\rho\, n_{\max} \le P(\tau) \le C$ for every trajectory. If τ is correct, $R(\tau) = 1 + P(\tau) \ge 1 - \rho\, n_{\max}$. If τ is wrong or unanswered, $r_{\text{out}}(\tau) = 0$ forces $v_i = 0$ for every call by Equation 5, so the cashback is zero and $R(\tau) = -\rho\, n(\tau) \in [-\rho\, n_{\max}, 0]$. For a correct τ_{c} and a wrong or unanswered τ_{w}, $R(\tau_{\text{c}}) - R(\tau_{\text{w}}) \ge 1 - \rho\, n_{\max} + \rho\, n(\tau_{\text{w}}) \ge 1 - \rho\, n_{\max}$, which is positive by (A2) and completes the proof. □

This says the price shapes the ranking within a correctness class and never across one: however many calls either trajectory made, a correct one stays above a wrong one. The margin is linear in the budget, $1 - \rho\, n_{\max}$, so the rent and the budget are set together: at our constants the margin is 0.7, at $\rho = 0.1$ it falls to 0.4, the arm of Table 14, and at $\rho\, n_{\max} = 1$ it vanishes. The cap $C = 0.7$ is chosen to equal the margin, so a correct trajectory's reward lies in $[0.7, 1.7]$ and cashback can never exceed what the outcome guarantees.

A.5 Chance verification does not pay

Verification decides on an estimate. $\bar{U}_i^{\mathrm{rand}}$ averages $K = 3$ patch scores, so q_i is a noisy estimate of the value $q_i^\star$ that would use the expected patch score μ_i of the fixed call in place of $\bar{U}_i^{\mathrm{rand}}$. Since $z \mapsto \max\{w, z\}$ is 1-Lipschitz, Equation 8 gives $|q_i - q_i^\star| \le |\bar{U}_i^{\mathrm{rand}} - \mu_i|$, and for patch scores in $[0, 1]$ that are independent given the call, Hoeffding's inequality bounds $\Pr(|q_i - q_i^\star| \ge t)$ by $2e^{-2Kt^2}$, which exceeds one at $K = 3$ and $t = \varepsilon$. At these values the bound is vacuous and does not justify the deadzone, which is calibrated on null draws instead, Appendix C. Noise also pays asymmetrically. Suppose a call's true value is zero and its estimate errs by $+0.1$ or -0.1 with equal probability; if it were paid whenever the estimate cleared $\varepsilon = 0.05$, the positive error would earn $\gamma \cdot 0.05 = 0.025$ and the negative one nothing, an expected cashback of 0.0125 per call for a crop that is worth nothing. With rent, the same call expects $\frac{1}{2}(0.025) - \frac{1}{2}(0.05) = -0.0125$. Proposition 3 states the general condition.

Proposition 3 *(restated). Let X_i be the uncapped price of an executed call, $X_i = \gamma(q_i - \varepsilon)$ if $v_i = 1$ and $X_i = -\rho$ otherwise, and let $A_i = \{q_i^\star \le \varepsilon\}$ with $\Pr(A_i) > 0$. Under (A1), if $\Pr(v_i = 1 \mid A_i) \le \alpha \in [0, 1)$, then $\mathbb{E}[X_i \mid A_i] \le \alpha\gamma(1 - \varepsilon) - (1 - \alpha)\rho$, which is negative under (A3).*

Proof. The uncapped trajectory price is $\sum_i X_i$, since the cashback sum runs over verified calls and the rent over the others. Condition on A_i throughout and let $p = \Pr(v_i = 1 \mid A_i) \le \alpha$. On the verified event, $q_i \le 1$ by (A1), so $X_i \le \gamma(1 - \varepsilon)$; on the unverified event, $X_i = -\rho$. Hence

$$\mathbb{E}[X_i \mid A_i] \le p\,\gamma(1 - \varepsilon) - (1 - p)\,\rho = p\,\big(\gamma(1 - \varepsilon) + \rho\big) - \rho \le \alpha\gamma(1 - \varepsilon) - (1 - \alpha)\rho,$$

where the last step uses $p \le \alpha$ and $\gamma(1 - \varepsilon) + \rho > 0$. The bound is negative if and only if $\alpha(\gamma(1 - \varepsilon) + \rho) < \rho$, that is, $\alpha < \rho/(\rho + \gamma(1 - \varepsilon))$, equivalently $\rho > \frac{\alpha}{1-\alpha}\gamma(1 - \varepsilon)$, which is (A3). At our constants the threshold is $0.05/(0.05 + 0.5 \cdot 0.95) = 0.05/0.525 \approx 9.5\%$, which completes the proof. □

Read as a condition on the rent, (A3) is $\rho > \frac{\alpha}{1-\alpha}\gamma(1 - \varepsilon)$: the rent must exceed the cashback that chance verification can collect. The bound is a worst case that credits every chance verification with the largest possible cashback; at the measured chance rate of 2.5% it gives $\mathbb{E}[X_i] \le 0.025 \cdot 0.475 - 0.975 \cdot 0.05 \approx -0.037$. Capping only lowers this: with S_{i-1} the pre-cap cashback of the calls before i, the allocated capped price $Y_i = \min\{C, S_{i-1} + b_i\} - \min\{C, S_{i-1}\} - \rho(1 - v_i)$ satisfies $Y_i \le X_i$ by Lemma 1(c) and $\sum_i Y_i = P(\tau)$, so the bound on $\mathbb{E}[X_i]$ also bounds $\mathbb{E}[Y_i]$. The premise is a property of the verifier rather than of the policy. For a fixed call distribution, verification implies clearing ε, so the chance rate of clearing ε bounds the verification rate of a call with no evidence; the null calibration estimates that rate on the cold start, with sampling error, and does not track the call distribution as training moves it. Training does not enforce the condition, and a rate above $2/21$, about 9.5%, would void the guarantee rather than reverse it, since the bound is an upper bound on the expectation.

A.6 Normalization and the price scale

GRPO divides each group's reward deviations by their standard deviation. When the price is part of a single reward and the outcome is unanimous, that standard deviation is the price's own spread, and dividing by it erases the units the rent was set in. Proposition 4 makes this exact in the cleanest case, and the corollary shows that the dual channel keeps Proposition 2 at the level of advantages.

Proposition 4 *(restated). Under (A4), in a group whose rollouts are all correct, carry no verified call, and differ in their unverified call counts n_j, the single-channel advantage is $A_j = -(n_j - \bar{n})/\mathrm{std}(n)$, independent of ρ, whereas Equation 7 gives $\widehat{A}_j = -\rho\,(n_j - \bar{n})$.*

Proof. With $n_j = n(\tau_j)$, every rollout has $O_j = 1$, $P_j = -\rho\, n_j$, and $R_j = 1 - \rho\, n_j$, so $R_j - \bar{R} = -\rho\,(n_j - \bar{n})$ and $\mathrm{std}(R) = \rho\,\mathrm{std}(n)$, which is positive because the counts differ. The group is therefore retained by the single-channel arm, and $A_j = (R_j - \bar{R})/\mathrm{std}(R) = -(n_j - \bar{n})/\mathrm{std}(n)$: the factor ρ cancels. In Equation 7, $O_j - \bar{O} = 0$ and the price channel is active because the group contains correct rollouts, so $\widehat{A}_j = 0/\sigma_{\min} + (P_j - \bar{P}) = -\rho\,(n_j - \bar{n})$, which completes the proof. □

In the single channel, a rent of 0.05 and a rent of 0.5 produce the same update in such a group: the policy is told which rollout called more, never by how much that cost. In the all-correct groups the proposition covers, with no verified call, every reward difference comes from unverified call counts, so these are the groups in which the rent alone shapes the policy. In mixed groups the scale is lost in the other direction: single-channel normalization divides a price difference by $\mathrm{std}(O + P)$, which depends on both channels and whose outcome part alone lies between 0.35 and 0.54, the worked example of Appendix C. The single-channel arm of Table 3 shows the consequence, 0.78 calls per question and an average of 64.5, below the cold start. The dual channel keeps the price in its own units in every group that contains a correct rollout, not only in unanimous ones: two rollouts with the same outcome have equal outcome terms, so $\widehat{A}_j - \widehat{A}_k = P_j - P_k$, and with otherwise equal prices one extra unverified call lowers their advantage difference by exactly ρ. Each statement rests on one implementation condition in (A4): the single-channel cancellation needs a denominator with no additive floor, since a floor η would turn it into an approximation valid only while $\rho\,\mathrm{std}(n) \gg \eta$, and the dual-channel identity needs an active, unnormalized price channel; the rule of Section 3.5 for groups with no correct rollout lies outside both statements. The outcome channel still dominates the price channel.

Corollary 1 (Outcome gap of the dual-channel advantage)**.** *Under (A2) and (A4), in a group with both outcomes, a correct rollout τ_j and a wrong or unanswered rollout τ_ℓ satisfy $\widehat{A}_j - \widehat{A}_\ell \geq 1/\max\{\mathrm{std}(O), \sigma_{\min}\} - \rho\, n_{\max}$, which is at least $\sqrt{7/2} - 0.3 \approx 1.57$ at our constants.*

Proof. Let $d = \max\{\mathrm{std}(O), \sigma_{\min}\}$. Since $O_j = 1$ and $O_\ell = 0$, the group means cancel in the difference and $\widehat{A}_j - \widehat{A}_\ell = 1/d + P_j - P_\ell \geq 1/d - \rho\, n_{\max}$, using $P_j \geq -\rho\, n_{\max}$ and $P_\ell \leq 0$ from the proof of Proposition 2. With k correct rollouts of eight, $\mathrm{std}(O)^2 = k(8-k)/56 \leq 16/56$, so $d \leq \sqrt{16/56} \approx 0.535$ and $1/d - \rho\, n_{\max} \geq \sqrt{56/16} - 0.3 = \sqrt{7/2} - 0.3$, which completes the proof. □

The gap stays positive as long as $\rho\, n_{\max} < 1/\max\{\mathrm{std}(O), \sigma_{\min}\}$; uniformly over mixed groups of eight this holds whenever $\rho\, n_{\max} < \sqrt{7/2} \approx 1.87$, a weaker condition than (A2), and the outcome channel, not the price channel, decides the order of a correct and a wrong rollout.

A.7 Examples

Table 5 collects the numbers the results take at the constants of Appendix C.

Table 5: The results at the paper's constants, $\gamma = 0.5$, $\varepsilon = 0.05$, $\rho = 0.05$, $C = 0.7$, $n_{\max} = 6$, $\sigma_{\min} = 0.15$, $G = 8$.

Quantity	Expression	Value
Outcome margin, Proposition 2	$1 - \rho\, n_{\max}$	0.70
Reward range of a correct trajectory	$[1 - \rho\, n_{\max},\ 1 + C]$	$[0.70, 1.70]$
Verifier reliability threshold, (A3)	$\rho/(\rho + \gamma(1-\varepsilon))$	9.5%
Measured chance rate of clearing ε	V*, HR-4K, free-form	2.4%, 2.5%, 3.0%
Worst-case expected price of a chance call at $\alpha = 2.5\%$	$\alpha\gamma(1-\varepsilon) - (1-\alpha)\rho$	-0.037
Advantage gap, Corollary 1	$\sqrt{7/2} - \rho\, n_{\max}$	1.57
Pairwise advantage decrement per extra unverified call	dual ρ; single $1/\mathrm{std}(n)$	0.05; ρ cancels

A.8 Scope and assumptions

Three assumptions carry the analysis, and we state them plainly. First, all scores are the policy's own gold-answer probabilities under a fixed elicitation, so every statement is about the reward as

computed, not about what the task requires or what the network attends to. Second, Proposition 3 is an expectation statement whose premise is a conditional false-verification rate; the null calibration replaces the real observation by one random patch and keeps the call's own D_i, so it is a proxy for that rate rather than a measurement of it, it is taken on the cold start rather than controlled during training, and the statement says nothing about the variance of the price or about finite-sample behaviour. Third, the free-form score is a mean log-probability, which has no finite range in general; the statements that use (A1) apply to free-form calls only with an explicit bound B on the score range, with $\gamma(1-\varepsilon)$ replaced by $\gamma(B-\varepsilon)$, and the deadzone for free-form calls is calibrated separately. We make no claim that the counterfactual values identify a causal effect of the pixels on the answer, and no convergence claim for the optimizer: the guarantees concern the reward and the advantage that training receives, and the experiments of Section 4 are the evidence for what training does with them.

B Data Curation Details

Cold-start trajectories are removed when a crop is degenerate or repeats an earlier crop in the same trajectory, when the reasoning or the tool call cannot be parsed, or when rollouts of the same question end in different final answers; 6,960 of the 7,267 released trajectories remain. The RL pool is built once, before any RL run, in five steps.

1. Source pool. The VisualProbe training split, 4,000 prompts, and a sample of 24,000 prompts from the DeepEyes training data, stratified over its three sub-sources with 16,800 from visual search, 3,600 from ArxivQA, and 3,600 from ThinkLite-VL, 28,000 prompts in total.
2. Near-duplicate removal. Prompts that share both a perceptual image hash and a question are collapsed to one, leaving 26,600.
3. Decontamination. A prompt is removed when its image lies within perceptual-hash Hamming distance 5 of any image in V*, HR-Bench-4K, HR-Bench-8K, MME-RealWorld, MMStar, ChartQA, or BLINK, leaving 26,200.
4. Format filter. Prompts whose reference answer is missing or cannot be parsed into an option letter or a short string are removed, leaving 25,700.
5. Difficulty filter. Each remaining prompt is sampled eight times from the SFT checkpoint under the training decoding settings and bucketed by the number of correct samples under the correctness rule of Section 4.1. Prompts with no correct sample are removed, prompts with one to seven correct samples are kept, and a random half of the prompts with eight correct samples is kept, so that groups with zero outcome variance do not dominate the pool. The final pool has 16,000 prompts.

Every RL arm trains on the same list.

C Training Configuration

Reward constants. $\gamma = 0.5$, $\varepsilon = 0.05$ for multiple-choice calls and $\varepsilon_{\text{free}} = 0.44$ for free-form calls, $\rho = 0.05$, $C = 0.7$, $K = 3$, $n_{\max} = 6$, and $\sigma_{\min} = 0.15$. Every arm uses the budget $n_{\max} = 6$. $\rho = 0.05$ gives $\rho\, n_{\max} = 0.3 < 1$, the condition of Proposition 2, and $C = 1 - \rho\, n_{\max} = 0.7$ caps cashback at the separation that the proposition guarantees. $\varepsilon = 0.05$ is the first grid value of the null calibration at which the false-verification rate falls under 3%, 2.39% on V* and 2.51% on HR-Bench-4K. $K = 3$ is the smallest number of patches beyond which the used label no longer moves, Table 13: agreement with $K = 3$ is 94.7% at $K = 1$ and 97.5% at $K = 2$, and $K = 5$ changes 2% of the labels. $\gamma = 0.5$ makes cashback comparable to the rent: cold-start verified calls at the median margin $q = 0.12$ and the upper quartile 0.23 receive 0.03 and 0.09 against a rent of 0.05, and six verified calls reach the cap only at a mean margin above 0.28. $\sigma_{\min} = 0.15$ lies below 0.35, the smallest non-zero outcome standard deviation in a group of eight, so it acts only in unanimous groups. Table 14 sweeps γ and ρ.

Random-patch variance. Given the prefix and the call, if the K random-patch scores are independent draws from the reference distribution, the Monte Carlo part of the variance of E_i is their variance divided by K; the real-crop score and the prefix are fixed, so K controls the random-patch

side only. Table 13 shows the used label stabilizing in K: one patch reproduces 94.7% of the labels of $K = 3$, two patches 97.5%, and five patches 97.8%, so beyond three the extra score evaluations buy almost no change in which calls verify.

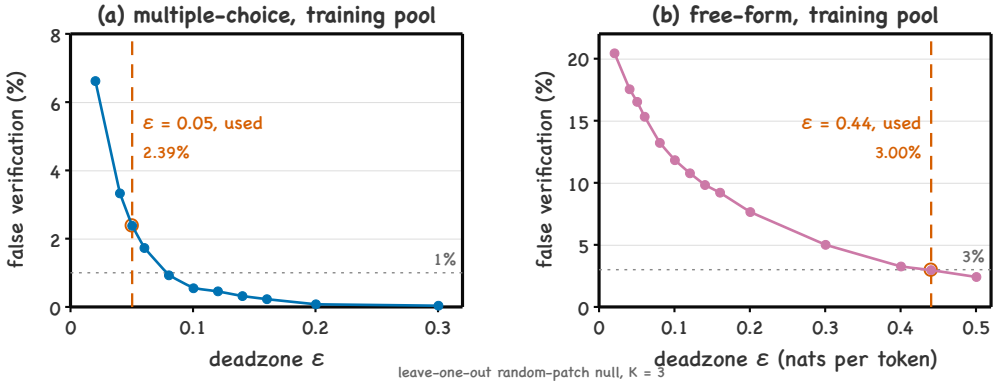


Figure 6: Null calibration of ε with $K = 3$. Leave-one-out null: each random-patch reading stands in for the real observation and is contrasted with the mean of the other patches, so its evidence has zero expectation; the call's own D is kept. Curves give the share of null draws with $\min(E, D) > \varepsilon$. (a) Multiple-choice, 500 training-pool questions probed from the cold start; the V* and HR-Bench-4K curves coincide with it at every grid value, 2.39% and 2.51% at $\varepsilon = 0.05$. (b) Free-form, 500 training-pool questions scored by the mean log-probability per token of the reference; the first grid value under 3% is $\varepsilon_{\text{free}} = 0.44$. The expected payment of a null draw is negative at every grid value in both panels.

Calibration of the deadzone. The deadzone was set before training from draws that use random patches only. For each probed call, each random-patch score in turn stands in for the real observation and is contrasted with the mean of the remaining patches, a leave-one-out null whose evidence has zero expectation when the patches are exchangeable, while the call's own D_i is kept. A null draw is falsely verified when its $\min(E, D)$ exceeds the candidate ε. Probes are scored with the parameters that generated the rollout, in evaluation mode, and are cached by prefix and call index. Every call's two values, verification status, and billing reason are logged, so that an unparsable call cannot silently become a free trajectory. Wrong or unanswered trajectories run no probes.

The verification threshold $\varepsilon = 0.05$ was fixed before training from the null calibration in Figure 6(a) on 500 multiple-choice questions of the training pool, which gives the share of zero-evidence draws that verify at each grid value; the same calibration on the cold-start probes of V* and HR-Bench-4K reproduces the curve, 2.39% and 2.51% at $\varepsilon = 0.05$, and the expected payment of a zero-evidence draw is negative at every ε in the grid. The free-form deadzone was calibrated by the same procedure on 500 free-form questions of the training pool, 1,389 calls and 4,167 null draws, with the score of Appendix D, the mean log-probability per token of the full reference under teacher forcing, and the answer-now anchor. In these units the multiple-choice threshold is far too low: the false-verification rate is 16.6% at $\varepsilon = 0.05$ and 11.9% at 0.10, and the first grid value under 3% is $\varepsilon_{\text{free}} = 0.44$ nats per token, 125 of 4,167 draws, which is the deadzone used for free-form calls, Figure 6(b). The expected payment of a null draw is negative at every grid value up to 0.30, from -0.018 at 0.02 to -0.039 at 0.30.

Cold-start training. Qwen2.5-VL-7B-Instruct is fine-tuned on the 6,960 cold-start trajectories for 3 epochs with a learning rate of 10^{-5} and a global batch of 32, training all parameters except the vision encoder and the projector, at a maximum input-pixel budget of 2M.

RL training. GRPO runs for three epochs over the pool of Appendix B, 188 steps in total, with 256 prompts per step, eight samples per prompt, mini batches of 32, a constant learning rate of 10^{-6}, and no KL or entropy regularization. Rollouts use temperature 1.0 and at most six tool interactions, on eight NVIDIA H100 80 GB GPUs.

Training cost. Both RL arms score every eligible call with the probe, the outcome-only arm for logging only, so their step times are directly comparable: the mean wall-clock time of a training step

is 2,972 s for CounterCredit and 2,881 s for outcome-only GRPO on eight H100 GPUs, a ratio of 1.03, so the reward computation beyond probing is negligible. A step probes about 2,200 eligible calls at $K + 2$ score evaluations each, one forward pass without generation per evaluation, with prefixes and calls cached. A run with the probes disabled averages 2,320 s per step, so probing adds 24% and CounterCredit costs 1.28 times a standard outcome-only pipeline; $K = 1$ would cut the probe cost by two fifths at the label agreement of Table 13. Inference carries no probe cost: the probes run only during training, and the latencies of Table 2 are measured without them.

Table 6 lists the ways a policy could farm credit under the reward of Section 3.4 and the component that blocks each of them in isolation.

Table 6: Ways to farm credit and the component that blocks each. Interactions between behaviours and gaming of the scoring policy itself are outside this table.

Behaviour	**What the policy does**	**How credit is constrained**
Text laundering	names the answer in the call's arguments	when the call text shifts the real and every random-patch score equally, that common shift cancels in E_i (Lemma 1)
Performative call	the branch changes the answer, the pixels do not	$E_i \approx 0$ fails $\min(E_i, D_i) > \varepsilon$
Unnecessary look	informative pixels after the answer was already settled	$D_i \leq \varepsilon$ fails $\min(E_i, D_i) > \varepsilon$
Looping, malformed calls	keeps calling, or never answers	never verified; each executed call pays ρ

Algorithm 1 lists the reward and advantage computation of Section 3.5 for one prompt group; the constants are those of Section 3.4. The worked example behind the dual-channel advantage, with the sample standard deviation the trainer uses: in a group of eight rollouts with seven correct and one wrong, the outcome standard deviation is 0.35, so a price difference of 0.05 between two otherwise identical correct rollouts becomes about 0.14 outcome-normalized units under standard GRPO, nearly three times its face value; with six correct of eight it is about 0.11, and with eight correct of eight the standard deviation is zero and the price is the only signal left; with no correct rollout the price advantage is set to zero. Standard GRPO divides by the standard deviation of the total reward, which differs from that of the outcome by the price variance, a few hundredths, so the outcome standard deviation stands in for it here. With eight rollouts per group and binary outcomes, that standard deviation takes only five values: zero in a unanimous group and between 0.35 and 0.54 otherwise, so standard GRPO leaves the price without a fixed scale in every unanimous group and scales a rent of 0.05 to between 0.09 and 0.14 outcome units in every other group; Proposition 4 gives the exact form of the all-correct case. The single-channel arm of Table 3 normalizes $R(\tau)$ by its group standard deviation and drops groups whose total reward is constant across the eight rollouts; in a unanimous group with unequal prices, the price differences are therefore rescaled to unit variance.

D Evaluation Protocol Details

All benchmarks are loaded and scored through VLMEvalKit (Duan et al., 2024) with its default per-benchmark rules: option-letter extraction for the multiple-choice benchmarks, relaxed accuracy for ChartQA, and the kit's answer matching for free-form questions. Each model runs its released prompts and tool loop under greedy single-pass decoding; our model uses the same tool budget, image budget, and prompts as in training. Closed models are queried through their public APIs with the standard VLMEvalKit prompts, temperature 0, and no tool loop.

Latency is the wall-clock time of one agent rollout per question at batch size 1 on one NVIDIA H20 with vLLM in BF16, measured from the first model generation to the final answer or stop. It includes every generation turn, every crop or Python tool execution, and the multi-turn interaction. It excludes model loading, dataset reading and initial image decoding, the offline counterfactual probes, and the answer judge for free-form responses, which runs after timing ends.

Judge check. We sampled 500 free-form answers from the training pool that reached the GPT-4o judge and labelled each by hand against the reference. The judge agreed with the label on 497, a 99.4% agreement; the three disagreements were answers that named the correct entity with an extra qualifier the judge rejected.

Algorithm 1 CounterCredit reward and advantage for one prompt group

Require: rollouts $\tau_1, \ldots, \tau_G$; constants $\varepsilon, \rho, \gamma, C, K, \sigma_{\min}$

for $j = 1, \ldots, G$ **do**
  $O_j \leftarrow \text{judge}(\tau_j)$; $P_j \leftarrow 0$; $b_j \leftarrow 0$
  for each executed call i of τ_j **do**
    if $O_j = 0$, or τ_j is unanswered, or the call did not return exactly one image, or no distinct patch exists **then**
      $P_j \leftarrow P_j - \rho$; **continue**
    end if
    $U_i^{\text{real}} \leftarrow \text{score}(h_i, c_i, o_i)$; $U_i^{\text{now}} \leftarrow \text{score}(h_i)$; $\bar{U}_i^{\text{rand}} \leftarrow \frac{1}{K}\sum_k \text{score}(h_i, c_i, \tilde{o}_{ik})$
    $E \leftarrow U_i^{\text{real}} - \bar{U}_i^{\text{rand}}$; $D \leftarrow U_i^{\text{real}} - U_i^{\text{now}}$; $q \leftarrow \min(E, D)$
    if E, D finite and $q > \varepsilon$ **then**
      $b_j \leftarrow b_j + \gamma(q - \varepsilon)$
    else
      $P_j \leftarrow P_j - \rho$
    end if
  end for
  $P_j \leftarrow P_j + \min(C, b_j)$
end for
if $O_j = 0$ for all j **then**
  $P_j \leftarrow \overline{P}$ for all j
end if
$\widehat{A}_j \leftarrow (O_j - \overline{O})/\max\{\text{std}(O), \sigma_{\min}\} + (P_j - \overline{P})$ for all j

Free-form score. For a reference answer y with tokens $y_1, \ldots, y_T$, the score of a prefix p is $\frac{1}{T}\sum_{t=1}^{T} \log \pi_\theta(y_t \mid p, e, y_{<t})$, computed under teacher forcing with the assistant turn prefilled by the answer tag; y is the reference string of the dataset as released. E_i and D_i are differences of this score, so they are in nats per token, and the deadzone in these units is $\varepsilon_{\text{free}} = 0.44$, Appendix C.

Spurious-call rate. The code-based agents can perform operations beyond cropping; every executed call that returns an image is treated as a look and audited in the same way, since each such call raises the same two questions. For every agent, E_i and D_i are computed with the agent's own gold-answer score on its own prefixes, with the same $K = 3$ random same-size patches and the same deadzone $\varepsilon = 0.05$ as in training. For a call whose returned image is not a crop, the reference is random patches from I of the returned image's size; for a full-size return this is the image itself, so E_i then measures the effect of the transform alone. Random patches pass through the same processing as the real observation. Calls that return no image are excluded from the denominator.

E Prompts and Formats

Every arm trained in this paper uses the Mini-o3 chat format (Lai et al., 2026); the released baselines run in their own formats. A trajectory alternates `<think>` reasoning, a `<grounding>` call written as a bounding box, and the returned crop as an image turn; h_i ends with the `<think>` block before call i, c_i is the `<grounding>` block, and o_i is the image that follows it. The system prompt below is the only one used, for the cold start, for every RL arm, and at evaluation; the user turn holds the image followed by the question and its options, one option per line. Complete trajectories are given in Appendix F.7.

System prompt

```
You are a helpful assistant. Answer the user's question based on the
↪ image provided. Output your thinking process within the <think> and
↪  </think> tags. Whenever you find anything unclear, you can zoom in
↪  on a specific region in the given image to see more clearly by
↪ outputting <grounding>{"bbox_2d": [x0, y0, x1, y1], "source": "
↪ original_image"}</grounding>, where (x0, y0) and (x1, y1) are the
↪ top-left and bottom-right coordinates of the region that you want
```

```
↪ to zoom in, respectively (suppose the width and height of the image
↪  are 1.0), and 'source' refers to the image that you zoom in and
↪ could be either 'original_image' or 'observation_i'. Once the final
↪  answer is confirmed, put it within <answer> and </answer>.
```

A <grounding> call is executed by cropping the named source at the given box, and the crop is returned in a user turn of the following fixed form, where i is the zero-based call index.

Tool-return template

```
After the above Action {i}, here is  the zoom-in image (Observation {i
↪ +1}):
[image]
.
Continue your reasoning process inside <think> and </think>. If needed
↪ , you can continue to zoom in on the original image or any of the
↪ observations, by outputting <grounding> and </grounding> as before.
↪  If the final answer is confirmed, put your final answer inside <
↪ answer> and </answer>.
```

The probe of Section 3.1 appends the fixed string e below as a user turn to the prefix being scored, merged into the trailing user turn when the prefix already ends in one, and prefills the assistant turn with <answer> followed by a space, so that the first generated token is the option letter; free-form questions use the second string below, and the reference answer is then scored under teacher forcing, Appendix D. The gold-answer score is the probability of the gold letter renormalized over the option letters at that position. The same string and prefill are used for the answer-now, real-observation, and random-observation continuations.

Answer elicitation e for multiple-choice questions

```
Based on everything so far, answer the original question now. Reply
↪ with the option letter only.
```

Answer elicitation e for free-form questions

```
Based on everything so far, answer the original question now. Reply
↪ with the answer only.
```

F Additional Results

F.1 What each existing reward pays for

The outcome reward credits every executed call on a correct trajectory. The branch probe of TACO scores the committed answer, the argmax over option letters under a rule-based checker, with and without the tool branch, and credits a call when the branch turns a wrong answer right. The external judge follows CodeV's reward prompt verbatim, served by GPT-4o, scores each returned crop 1, 0.5, or 0.25, and credits it at 1. The self-judge follows the FaithEyes prompt verbatim, asking the policy itself whether the crop is helpful in a fresh context holding only the question and that crop, and credits it when the probability of a helpful verdict exceeds 0.5. Table 7 resolves Figure 1(b) by benchmark, over 2,594 scored calls: the branch probe has the highest share of paid calls passing both tests on every benchmark, and the three rewards that do not intervene trade places below it.

Common scorer. The spurious-call rates of Table 2 use each agent's own gold-answer score on its own prefix. To check that the ordering is not an artifact of the scorer, we re-scored the executed calls of CounterCredit and of all seven baselines on V^* with a single scorer, the SFT cold start, holding each agent's prefixes and returned crops fixed. CounterCredit keeps the lowest spurious-call rate under the common scorer, and on the same rollouts the ordering of the seven baselines under

Table 7: The reward audit of Figure 1(b) per benchmark, on the SFT checkpoint's trajectories. Paid: calls credited per 100 executed calls; of those, the share needed ($D_i > 0.05$), used ($E_i > 0.05$), and both. The last row pays every call and gives the base rates. CounterCredit is omitted because its verification is defined by the same two values.

	V*				HR-4K				HR-8K			
Reward	**Paid**	**Needed**	**Used**	**Both**	**Paid**	**Needed**	**Used**	**Both**	**Paid**	**Needed**	**Used**	**Both**
Outcome reward	40.2	17.4	18.6	12.8	39.2	23.2	22.5	12.4	33.7	31.1	24.0	14.6
Branch probe (TACO)	5.3	61.1	50.0	41.7	5.6	52.5	50.8	41.7	6.7	60.7	57.2	49.7
External judge	23.4	30.2	26.9	21.0	28.6	33.1	30.4	19.3	24.8	32.6	32.3	22.4
Self-judge	19.9	22.1	27.2	14.7	33.0	24.6	33.1	20.9	28.0	29.4	30.5	22.6
Every call paid	100	13.9	15.1	11.8	100	16.1	18.2	9.8	100	18.1	16.5	10.4

their own scorers and under the common scorer agree up to one adjacent swap, Pixel-Reasoner and PyVision-RL, with Spearman $\rho = 0.96$ between the two orderings. Absolute rates depend on the scorer, so only the ordering is compared across agents. A third scorer that took no part in training or in any metric of this paper, GPT-5.6 Terra (OpenAI, 2026), recomputes the score of Equation 1 on the same prefixes, calls, and observations, reading the gold option's log-probability from the interface and renormalizing it over the option letters, with structured outputs disabled so that the log-probabilities are returned. The share of calls whose real observation does not beat the mean of its three random patches under this reader runs from 20.0% for Thyme to 74.4% for DeepEyesV2, orders the agents at Spearman $\rho = 0.75$ against the common scorer, and again leaves CounterCredit the lowest. The check asks whether an outside reader gains from the returned pixels at the same state; it does not reproduce the spurious-call rate, which also requires D_i and the deadzone.

Answer-level counterfactual. The four regions are defined by probability contrasts, so we also test them with a different endpoint. Every eligible call of the seven baselines on V*, $1{,}835$ in all, is re-decoded with the real crop replaced by each of its three random patches in turn, $5{,}505$ draws, and we record whether the decoded answer changes. Table 8 gives the rates. Real calls change the answer in a third of draws, sixteen times the rate of waste. Performative calls change it at the rate of waste, 2.2% against 2.0%, although their decision value is positive: a reward that checks the decision value alone credits them, and their returned pixels move the answer as rarely as a wasted call's. Unnecessary calls sit in between, on the smallest of the four regions. The endpoint is new but the regions and the intervention are ours, so this is corroboration rather than an independent test, and a changed answer is not necessarily a corrected one.

Table 8: Answer-level counterfactual on V*, pooled over the seven baselines. Each executed call is re-decoded once per random patch; *changed* is the share of draws whose decoded answer differs from the one the real crop produced.

Region	**Changed (%)**
Real	33.3
Performative	2.2
Unnecessary	17.1
Waste	2.0

F.2 Behaviour behind Figure 5

Table 9 lists the accuracies plotted as gaps in Figure 5(a), on V* and on HR-Bench-4K. The two benchmarks agree on the reading: the real crop moves the two baselines by at most 1.1 points either way, while it is worth 6.8 points to CounterCredit on V* and 4.3 on HR-Bench-4K, three fifths of the distance from answering at once to the full rollout on the first and two thirds on the second. Table 10 lists the tercile values plotted in panels (b) and (c).

F.3 Generalization across base models

Table 11 gives both bases on all seven benchmarks; both rise on all seven, so the recipe does not trade general competence for tool use.

Table 9: Accuracy when the trajectory is stopped at its first tool call and the answer is elicited from the cached prefix, for the arms of Table 3. No call: the call is removed. Random: the call returns a random same-size patch, averaged over three patches. Real: the call returns its real crop. Full: the complete rollout.

Model	No call	Random	Real	Full
V*				
SFT (cold start)	75.4	73.8	75.4	71.7
Outcome-only GRPO	77.5	74.7	76.4	80.1
CounterCredit	78.5	74.3	85.3	89.5
HR-Bench-4K				
SFT (cold start)	71.6	68.9	71.9	70.4
Outcome-only GRPO	73.2	69.9	72.9	73.9
CounterCredit	73.5	70.5	77.8	80.2

Table 10: V* by the target's share of the image area, split into terciles at 0.031% and 0.110% of the image, about 64 questions each: accuracy (%) and calls per question for the same three arms.

	Smallest		Middle		Largest	
Model	**Acc.**	**Calls**	**Acc.**	**Calls**	**Acc.**	**Calls**
SFT (cold start)	65.1	3.33	70.3	2.84	79.7	1.88
Outcome-only GRPO	66.7	2.38	89.1	1.77	84.4	1.39
CounterCredit	82.5	2.19	92.2	1.70	93.8	1.45

F.4 Seed variation

Table 12 lists the three CounterCredit seeds behind the $\pm$ values of Table 1; the seven-benchmark average of each seed is computed from its own row, and the standard deviation is the sample standard deviation over the three.

F.5 Sensitivity of the evidence value

Table 13 varies the two design choices behind E_i on the cold-start audit calls, the number of random patches K and the reference distribution. Each row re-scores the same calls with the same prefixes and real crops and changes only the reference; used means $E_i > \varepsilon$, agreement is the share of calls whose used label matches the default, and r is the correlation of the re-scored E_i with the default.

The two benchmarks agree. Dropping two of the three patches keeps 94.7% and 94.8% of the used labels on V* and HR-Bench-4K, with correlations of 0.94 and 0.96 to the default, dropping one keeps 97.5% and 96.7% with 0.98 and 0.99, and adding two more keeps 97.8% and 97.1% with 0.99 on both, so further patches change few labels. The blurred real crop changes the most labels, followed by the gray patch: the blur keeps layout and colour, so a call whose evidence survives the blur is no longer counted as used, and the gray patch, which on average scores as a random patch does, disagrees with the default call by call. A patch that overlaps the real crop agrees with the default about as closely as leaving one patch out, so the label does not depend on the patches being far from the evidence.

F.6 Extended ablation

Table 14 adds one rent arm and four hyperparameter arms to Table 3, each trained from the same cold start with the same prompt order, seed, group size, and number of steps as the reference run.

Charging rent only on correct trajectories costs the most selectivity. A wrong trajectory then pays nothing for its calls, so the price channel ranks it above a correct trajectory that carries one unverified call, and both the call count and the spurious-call rate move toward the arm without rent. The four sweeps put the reference constants at the accuracy optimum, and every departure costs 1.5 to 3.1 points on V*. The two arms that share the ratio of γ to ρ, $\gamma = 0.25$ and $\rho = 0.1$, behave alike: the ratio fixes how the price channel orders the rollouts of a group, and the two arms differ only in the weight of that channel against the outcome channel; both cut calls and raise the spurious share.

Table 11: Two base models on all seven benchmarks (accuracy %). Base rows answer directly; + rows apply our cold start and RL; no reward ablation was run on Qwen3-VL-8B.

Base / Method	V*	HR-4K	HR-8K	MME-RW	MMStar	ChartQA	BLINK	Avg.
Qwen2.5-VL-7B	77.0	69.1	65.6	58.0	64.4	83.9	56.2	67.7
+ CounterCredit	89.5	80.2	76.4	67.7	68.9	87.1	62.8	76.1
Qwen3-VL-8B	85.9	79.2	74.3	60.7	70.5	88.4	68.9	75.4
+ CounterCredit	91.6	85.4	84.0	70.1	72.2	90.0	72.3	80.8

Table 12: Three training seeds of CounterCredit, evaluated identically. Seed 0 is the run reported everywhere else in the paper.

Seed	V*	HR-4K	HR-8K	MME-RW	MMStar	ChartQA	BLINK	Avg.	Calls	Spur. %
0	89.5	80.2	76.4	67.7	68.9	87.1	62.8	76.1	1.78	31.4
1	90.1	80.9	77.2	68.2	68.5	86.8	63.4	76.4	1.84	33.3
2	88.5	79.4	75.4	67.2	69.2	87.4	62.1	75.6	1.72	29.5
mean	89.4	80.2	76.3	67.7	68.9	87.1	62.8	76.0	1.78	31.4
sample SD	0.81	0.75	0.90	0.50	0.35	0.30	0.65	0.40	0.06	1.90

Doubling γ or lowering ρ to 0.02 moves the ratio the other way, and both arms add calls and raise the spurious share further.

F.7 Case study

Four calls of the cold-start checkpoint on V*, one per region of Figure 3; Figure 7 shows where all of its calls fall. Each box shows the original image with the real crop in orange and the three random same-size patches in dashed blue, then the five observations the probe scores with their gold-answer scores U, the no-call score above the original image, and the trajectory verbatim with the tool-return turns abbreviated. In each box, h_i is everything before the assistant turn that issues the call, c_i is the orange `<grounding>` block, and o_i is the crop in the tool turn that follows.

Most calls sit within a few hundredths of the origin, where neither test separates them from noise, and the real region thins out along the diagonal as both values grow together. Performative calls form a narrow band just left of the vertical axis, where random patches score as well as the real crop, and unnecessary calls a band just below the horizontal axis, where the crop beats random patches but not answering at once; waste spreads down the diagonal, calls that hurt both ways.

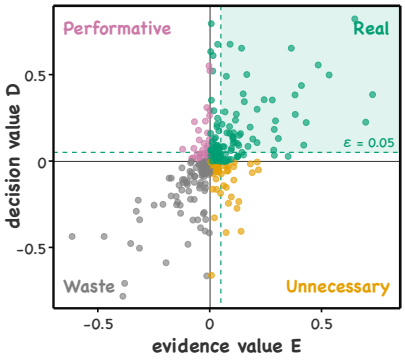


Figure 7: Calls of the cold-start checkpoint on V*, the first six per trajectory, in the plane of Figure 3. Dashed lines mark ε; the shaded region is where a call on a correct trajectory verifies.

Case 1, real: needed and used. A real call. Without the call the policy leans to the wrong side; the returned crop shows the cart to the left of the dog and moves the gold-answer score from 0.29 to 0.85, while three random same-size patches move it only to 0.37. The call is verified, $D_i = 0.56$ and $E_i = 0.48$, and earns cashback.

Table 13: Sensitivity of the evidence value to the number of random patches and to the reference distribution, on the cold-start audit calls. Default: $K = 3$ same-size random patches. Agreement: share of calls with the same used label as the default. r: Pearson correlation between the re-scored E_i and the default E_i. $K = 1$ and $K = 2$ are averaged over the three choices of the retained patch or pair; $K = 5$ adds two fresh patches to the stored three. Blurred real crop: the real crop after Gaussian blur, at its own size. Shifted patch: the stored random patches that overlap the real crop with intersection over union between 0 and 0.5, available for a subset of calls.

	V*			HR-4K		
Reference	**Used %**	**Agreement %**	r	**Used %**	**Agreement %**	r
Default, $K = 3$ random patches	15.1	100	1.00	18.2	100	1.00
$K = 1$	16.4	94.7	0.94	18.5	94.8	0.96
$K = 2$, leave one out	15.7	97.5	0.98	18.6	96.7	0.99
$K = 5$	14.9	97.8	0.99	18.1	97.1	0.99
Gray patch	13.7	90.0	0.84	14.8	90.0	0.92
Blurred real crop	10.9	88.5	0.79	13.5	89.5	0.87
Shifted patch	17.4	97.5	0.97	18.5	95.9	0.96

Table 14: Extended ablation: accuracy (%), calls per question, and spurious-call rate (%) on V* and HR-Bench-4K. Reference: the CounterCredit run of Table 3. The rent arm changes where ρ is charged; the sweeps vary one constant with the others fixed at $\gamma = 0.5$, $\rho = 0.05$.

		Accuracy		Calls / question		Spurious %	
Block	**Arm**	**V***	**HR-4K**	**V***	**HR-4K**	**V***	**HR-4K**
Reference	CounterCredit	89.5	80.2	1.78	1.76	31.4	35.1
Rent	rent only on correct trajectories	87.4	78.5	2.27	2.23	44.1	47.2
Cashback γ	$\gamma = 0.25$	87.4	78.6	1.57	1.55	36.2	39.7
	$\gamma = 1.0$	88.0	79.1	2.05	2.02	38.4	41.8
Rent ρ	$\rho = 0.02$	86.4	76.2	2.20	2.17	42.3	45.4
	$\rho = 0.1$	86.9	78.4	1.50	1.48	34.7	38.1

Case 1 **real: needed and used** **V* question 145**



Question Is the shopping cart on the left or right side of the dog? A. left B. right Gold: A

Assistant

```
<think> The user is asking about the relative position of a shopping
↪ cart and a dog. I need to locate both objects in the image to
↪ answer the question. I can see a dog on the sidewalk on the right
↪ side of the image, near some parked cars and people. I can also see
↪  a shopping cart next to the dog. To confirm their positions
↪ relative to each other, I will zoom in on the area where they are
↪ located. This area is on the right side of the image, so I'll
↪ create a bounding box around that region. A good bounding box would
↪  be from approximately 60% to 100% of the image width and 60% to
```

```
↪ 90% of the image height. </think><grounding>{"bbox_2d":
↪ [0.6,0.6,1.0,0.9], "source": "original_image"}</grounding>
```

Tool [Observation 1: the real crop shown above, 900 by 450 pixels at (1350, 900) to (2250, 1350) of the original image; template in Appendix E]

Assistant

```
<think> In this zoomed-in view, I can clearly see the dog and the
↪ shopping cart. The dog is standing on the snow-covered sidewalk. To
↪  the left of the dog, there is a yellow shopping cart. Therefore,
↪ the shopping cart is on the left side of the dog. The question asks
↪  for the left or right side, and based on this observation, the
↪ shopping cart is on the left side of the dog. I can confidently
↪ answer the question now. </think><answer> A. left </answer>
```

Final answer A. left, correct. $U_{\text{now}} = 0.29$, $\bar{U}_{\text{rand}} = 0.37$, $U_{\text{real}} = 0.85$, $D_i = 0.56$, $E_i = 0.48$

Case 2, performative: needed but not used. A performative call. The crop raises the gold-answer score from 0.65 to 0.88, so the look was needed, but the three random patches raise it to 0.95: receiving any observation helps the policy commit, and the pixels of the real crop add nothing. $D_i = 0.23$, $E_i = -0.07$; the call pays rent, and a reward on the decision value alone would have credited it.

Case 2 performative: needed but not used V* question 130



Question Is the wristband on the left or right side of the dog? A. right B. left Gold: B

Assistant

```
<think> The user is asking about the location of a wristband relative
↪ to a dog in the image. First, I need to locate the dog. I can see a
↪  small dog on a paddleboard in the center of the image. The dog is
↪ wearing an orange life vest. Next, I need to find the wristband.
↪ There is a person standing on the paddleboard with the dog, and
↪ this person is wearing a black wristband on their left wrist. The
↪ dog is on the right side of the person's body. Therefore, the
↪ wristband is on the left side of the dog. To confirm this, I will
↪ zoom in on the person and the dog. The area of interest is in the
↪ center of the image, so I'll create a bounding box around them. </
↪ think><grounding>{"bbox_2d": [0.4,0.4,0.7,0.8], "source": "
↪ original_image"}</grounding>
```

Tool [Observation 1: the real crop shown above, 450 by 859 pixels at (600, 859) to (1050, 1718) of the original image; template in Appendix E]

Assistant

<think> In this zoomed-in view, I can clearly see the woman and the dog on the paddleboard. The woman is wearing a black wristband on her left wrist. The dog is on the paddleboard to the right of her legs. Therefore, the wristband is to the left of the dog. The question asks if the wristband is on the left or right side of the dog. Based on this observation, it is on the left side. The answer is B. </think><answer> B. left </answer>

Final answer B. left, correct. $U_{\text{now}} = 0.65$, $\bar{U}_{\text{rand}} = 0.95$, $U_{\text{real}} = 0.88$, $D_i = 0.23$, $E_i = -0.07$

Case 3, used but not needed. A call whose pixels matter but whose look was not needed. The policy already answers correctly from the full image, 0.97, and the real crop keeps it there, 0.97, while random patches pull it down to 0.76. $D_i = -0.004$ and $E_i = 0.21$; the call pays rent because it changed nothing.

Case 3 used but not needed V* question 20



Question What is the color of the shovel? A. yellow B. red C. blue D. black Gold: C

Assistant

<think> The user is asking for the color of a shovel. I need to first locate the shovel in the image. The image shows a snowy park with a playground. I can see a person standing near the playground equipment. This person might be holding a shovel. I will zoom in on the area around the person to get a better look. The person is located in the lower central part of the image. I will crop the region around this person to see if there is a shovel. The coordinates for the crop will be approximately [0.4, 0.7, 0.6, 0.9]. </think><grounding>{"bbox_2d": [0.4,0.7,0.6,0.9], "source": "original_image"}</grounding>

Tool [Observation 1: the real crop shown above, 450 by 300 pixels at (900, 1050) to (1350, 1350) of the original image; template in Appendix E]

Assistant

<think> The user wants to know the color of the shovel. In the previous step, I zoomed in on a person near the playground. In observation 1, I can see a person on the left side of the image. This person seems to be holding a blue object. This object could be a shovel. I need to zoom in further to confirm the object and its color. The person and the object are located in the bottom-left corner of observation 1. I will crop this area to get a clearer view. The coordinates for the crop will be approximately [0.0, 0.5, 0.4, 1.0]. </think><grounding>{"bbox_2d": [0.0,0.5,0.4,1.0], "source": "observation_1"}</grounding>

Tool [Observation 2: a 180 by 150 pixel crop of observation 1; template in Appendix E]

Assistant

```
<think> The user wants to know the color of the shovel. In the
↪ previous step, I zoomed in on a person near the playground. In
↪ observation 2, I can see a person on the left side of the image.
↪ This person is wearing a blue jacket and is holding a blue object.
↪ This object is a shovel. I can clearly see the color of the shovel
↪ is blue. The question asks for the color of the shovel, and the
↪ options are A. yellow, B. red, C. blue, D. black. The color of the
↪ shovel is blue, which matches option C. Therefore, I can provide
↪ the final answer. </think><answer> C. blue </answer>
```

Final answer C. blue, correct. $U_{\text{now}} = 0.97$, $\bar{U}_{\text{rand}} = 0.76$, $U_{\text{real}} = 0.97$, $D_i = -0.004$, $E_i = 0.21$

Case 4, neither needed nor used. A harmful call. The policy is right before it looks, 0.96, and wrong after the crop returns, 0.18; random patches leave it at 0.57. $D_i = -0.78$ and $E_i = -0.39$. The crop shows the boat without the man in green shorts, the reasoning talks itself out of the correct answer, and the trajectory ends wrong; the call pays rent, and an outcome reward would not have scored it at all.

Case 4 neither needed nor used V* question 142



Question Is the man with green shorts on the left or right side of the blue boat? A. right B. left
Gold: B

Assistant

```
<think> The user is asking about the relative position of a man with
↪ green shorts to a blue boat. First, I need to locate the blue boat
↪ in the image. I can see a blue boat in the middle of the water,
↪ with two men on it. The question asks if the man with green shorts
↪ is on the left or right side of this blue boat. I need to examine
↪ the people on the boat to determine the color of their shorts. The
↪ people on the boat are quite small, so I will need to zoom in to
↪ see the details clearly. I will crop the region containing the blue
↪  boat to get a better view. The boat is located in the center of
↪ the image, so I'll create a bounding box around it. A good starting
↪  point for the bounding box would be [0.4, 0.45, 0.6, 0.6] to focus
↪  on the boat and its occupants. </think><grounding>{"bbox_2d":
↪ [0.4,0.45,0.6,0.6], "source": "original_image"}</grounding>
```

Tool [Observation 1: a 450 by 225 pixel crop of original image; template in Appendix E]

Assistant

```
<think> I have successfully zoomed in on the blue boat. In this new
↪ observation, I can clearly see two men. The man on the left is
```

```
↪ wearing blue jeans. The man on the right is wearing a light-colored
↪  shirt and dark pants. The question asks about a man with green
↪ shorts. I do not see anyone wearing green shorts on this boat. It
↪ is possible that the user is mistaken about the color of the shorts,
↪  or there is another person not visible in this crop. However,
↪ based on the information available in this observation, I cannot
↪ confirm the presence of a man with green shorts. The man on the
↪ left is on the left side of the boat, and the man on the right is
↪ on the right side. Since the question asks for the side relative to
↪  the boat, and there is a man on the right, the answer would be "
↪ right". However, I cannot be certain about the color of the shorts.
↪  I will go back to the original image and try to get a better view
↪ of the people on the boat, perhaps by adjusting the bounding box to
↪  include more of the surrounding area which might give more context.
↪  </think><grounding>{"bbox_2d": [0.35,0.4,0.65,0.6], "source": "
↪ original_image"}</grounding>
```

Tool [Observation 2: the real crop shown above, 674 by 300 pixels at (788, 600) to (1462, 900) of the original image; template in Appendix E]

Assistant

```
<think> I have examined the two men on the blue boat. The man on the
↪ left is wearing blue jeans, and the man on the right is wearing
↪ dark pants. The question asks if there is a man with green shorts.
↪ I have carefully inspected the image, and there is no one wearing
↪ green shorts. It is possible the user is mistaken. However, the
↪ question asks for the position of the man with green shorts
↪ relative to the boat. Since there is no such person, the question
↪ is based on a false premise. I will assume the user is asking about
↪  the position of the man who *could be* on the right side of the
↪ boat. There is a man on the right side of the boat. Therefore, the
↪ man with green shorts (if he existed) would be on the right side of
↪  the boat. The answer is 'right'. I am confident in this answer as
↪ it is the only logical interpretation of the question given the
↪ visual evidence. </think><answer> right </answer>
```

Final answer right, wrong. $U_{\text{now}} = 0.96, \bar{U}_{\text{rand}} = 0.57, U_{\text{real}} = 0.18, D_i = -0.78, E_i = -0.39$

G Further Limitations

Per-call values do not sum to the value of the trajectory. The evidence value requires a matched null return, which we define only for crops. Verification weakens when the gold-answer score saturates, which also inflates the spurious-call rate of an agent scored by its own saturated distribution, although the common and outside scorers of Appendix F preserve the ordering of agents. Charging rent on unnecessary calls, informative pixels with $D_i \leq \varepsilon$ as in Case 3 of Appendix F.7, penalizes confirmatory looks; whether such looks aid robustness under distribution shift was not tested. Random patches are unscreened, so a patch that contains the target raises the reference mean and lowers E_i; the reference errs toward under-verification. Multiple-choice tasks use a renormalized probability and free-form tasks a log-probability, and the free-form deadzone is calibrated separately, Appendix C. Reward construction costs $K + 2$ score evaluations per eligible call on a correct trajectory, and the probes run only during training. Open questions are tools that return text, evidence accumulated across several calls, further base families, and longer training in which the policy could adapt to the probe.