REFLECTION REWARDS PAY FOR DEFERRAL: CAPABILITY-AWARE REWARDS FOR REFLECTIVE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-reflective reinforcement learning trains models to revise their answers and is typically judged by final-answer accuracy. However, a reward for correcting mistakes also shapes the answer being corrected. We identify *deferral*, a failure mode in which a model postpones its real answer until after reflection, so that competitive final accuracy conceals a degraded first answer. In a controlled setting, a reflection-trained model's first answer falls below chance while its final answer remains accurate. We trace this to the reward: when a trajectory-level reward values a correction above a preserved correct answer, it introduces a bias toward deferral that grows with the model's capability. Natural remedies do not resolve this. Rewarding the reflection or answer changes can also pay for deferral, and since a deferred first answer is predictably wrong, rewarding calibrated confidence reinforces it too. We therefore propose Uncertainty-Aware Self-reflective Reinforcement Learning (UASRL), which uses a per-prompt capability proxy estimated once before RL, beyond the policy's control, and favors correct first answers where capability is high while still rewarding correction. Added to an existing reflection reward, UASRL substantially improves the first-answer accuracy of that reward and reduces deferral while maintaining competitive final accuracy. Across two backbones and three benchmarks, including a held-out one, it outperforms all tested reflection methods on first answers. Our results expose first-answer degradation as a hidden cost of reflection rewards, derive design criteria for such rewards, and show that self-reflection should be evaluated by both its first and final answers.

1 INTRODUCTION

A correct final answer does not tell us whether a model corrected a mistake or merely postponed answering. Self-reflection lets language and vision-language models review and revise their responses (Madaan et al., 2023; Shinn et al., 2023), and methods such as SCoRe (Kumar et al., 2025), SRPO (Wan et al., 2025), and VL-Rethinker (Wang et al., 2025) train this behavior with reinforcement learning from verifiable rewards (Shao et al., 2024; Guo et al., 2025). Yet the reward for a better second answer also shapes the first, and final-answer accuracy alone cannot tell whether reflection improves a genuine attempt or compensates for a degraded one.

We observe this sharply in a controlled setting whose SFT data contain wrong-first reflection templates. On ScienceQA, a model trained with a reflection reward gives a first answer that is correct less often than uniform guessing, while its final answer stays above 80%; on almost every question, its first option is the one that cyclically follows its own final option. Figure 1(b,c) shows how this unfolds during training. We call this behavior *deferral*: the first answer serves as a placeholder, and the real answer is left to the second attempt. The first answer matters in its own right: it is what a system returns when generation stops early to save latency or cost, and what a user reads first.

Why would a reward reinforce deferral? Consider the *correction-incentive gap* $g = R(0,1) - R(1,1)$ between correcting an initially wrong answer and preserving an initially correct one (Figure 1a); the released SRPO implementation has $g > 0$, and we call this reward SRPO†. When the trajectory's reward also credits the first answer, deferral makes an otherwise correct response eligible for the larger correction reward. We show that the reward's preference for deferral is its effect on final accuracy plus a bias with the sign of g that grows with the model's capability, so deferral that merely postpones the answer raises reward without improving it. Deferral is thus a form of reward hacking (Skalse

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

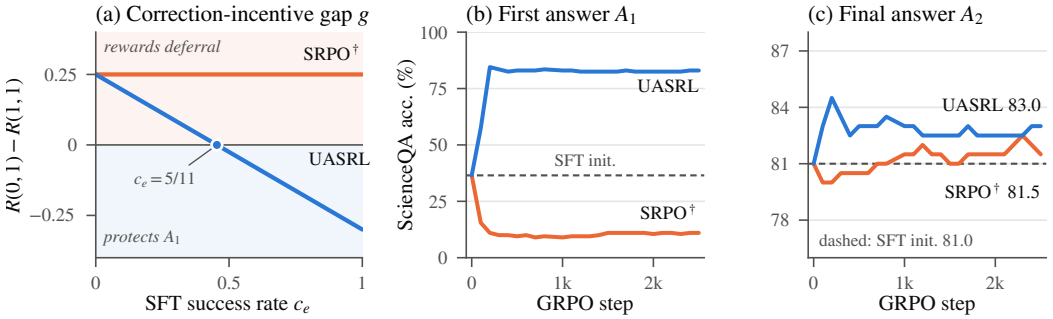


Figure 1: **Reflection rewards can teach a model to defer its answer.** (a) When a reward pays more for correcting a mistake than for getting it right the first time ($g > 0$), the model is incentivized to withhold its first answer; UASRL makes the gap negative where its capability estimate is high. (b, c) Accuracy on a ScienceQA subset during GRPO training. (b) Under SRPO†, the first answer collapses, while under UASRL it rises. (c) Final accuracy looks the same for both, so evaluating the final answer alone hides the failure.

et al., 2022; Pan et al., 2022) and a measurable instance of the superficial self-reflection noted in R1-style training (Liu et al., 2025). We study it mainly where the initialization makes it available, and give preliminary evidence that it also arises without templates.

The fix should reduce the gap where the model can answer while keeping correction rewarded, but the two natural ways of doing so fall short. Rewarding the reflection or answer changes can backfire: every deferral comes with a reflection that reports an error and a changed answer, so such rewards can pay for deferral. Protecting first answers the model is confident in also backfires: a deferred first answer is predictably wrong, so calibrated confidence is lowest exactly where deferral occurs, and rewarding it pays extra for deferral. What separates deferral from inability is not confidence in the answer given, but whether the model can answer when it tries.

We therefore propose **Uncertainty-Aware Self-reflective Reinforcement Learning (UASRL)**, which measures this once, before RL, as the SFT model's success rate on each sample: a fixed capability proxy that the policy cannot lower by deferring. Added to an existing reflection reward, it favors preserving correct first answers where this rate is high, while keeping correction preferable to persistent error everywhere. Across two backbones and three benchmarks, including a held-out one, UASRL improves first-answer accuracy over the shared initialization and all tested reflection methods while keeping final accuracy competitive, and it substantially reduces deferral. Our contributions are:

- We identify *deferral*, a failure mode of self-reflective RL in which the first answer degrades while final accuracy stays competitive, and show that evaluating the final answer alone hides it.
- We show that a trajectory-level reward with a positive correction-incentive gap adds a deferral bias that grows with capability, that rewarding reflections, answer changes, or calibrated confidence does not remove this bias, and derive design criteria for rewards that do.
- We propose UASRL, which adds a capability estimate fixed before RL to an existing reflection reward to favor correct first answers, and show across two backbones and three benchmarks that it improves first-answer accuracy and reduces deferral while keeping final accuracy competitive.

2 Related Work

Self-correction and reflection RL. Prompting-based self-correction (Madaan et al., 2023; Shinn et al., 2023) often fails without external signals (Huang et al., 2024; Kamoi et al., 2024), motivating training-based methods: supervised correctors (Welleck et al., 2023), fine-tuning on improvement trajectories (Qu et al., 2024), and RL objectives that reward verification and correction (Ma et al., 2025; Xiong et al., 2025) or the reflection itself (Wang, 2025; Jiang et al., 2026), building on RL for reasoning (Shao et al., 2024; Guo et al., 2025). Reflect-Retry-Reward (Bensal et al., 2025) and experiential RL (Shi et al., 2026) reinforce a second attempt that succeeds after a failed first one, but credit only the reflection or the retry. For VLMs, SRPO (Wan et al., 2025) trains reflection with

reflection-aware GRPO, VL-Rethinker (Wang et al., 2025) adds selective replay and forced rethinking, Sherlock (Ding & Zhang, 2025) and Octopus (Ding et al., 2026) enrich correction signals, and Wang et al. (2026) distinguish superficial from effective reflection, while Zhu et al. (2026) probe and steer self-reflection through directions in activation space. These methods improve what the model does during reflection; we study how the reward treats the answer that precedes it. SVR-R1 (Wu et al., 2026) uses an outcome-only reward, which has $g = 0$ and hence no deferral bias. SCoRe (Kumar et al., 2025) guards against the opposite failure, a second attempt that copies the first; its shaped objective has a positive gap, but it credits each turn only with its own reward, so the direct incentive we analyze does not apply. Multi-turn RL more broadly assigns credit per turn through learned values (Zhou et al., 2024) or turn-level rewards (Wei et al., 2025); we show that when the first answer instead shares the trajectory's reward, the gap shapes how it is produced.

Reward hacking. Policies optimized against imperfect rewards find behaviors that raise reward without the intended improvement (Skalse et al., 2022; Pan et al., 2022; Gao et al., 2023), including in reasoning models (Baker et al., 2025). Deferral is such a behavior with a specific structure: its incentive scales with capability, and when it merely postpones the answer, final-accuracy evaluation cannot detect it. It resembles sandbagging, deliberate underperformance by a capable model (van der Weij et al., 2024), but it is not an explicit objective: RL amplifies it because the reward pays for it.

Confidence and uncertainty. A line of work trains or evaluates stated confidence, verbalized or token-based (Guo et al., 2017; Kadavath et al., 2022; Tian et al., 2023; Xiong et al., 2024; Zhang et al., 2024a; Stangel et al., 2026; Li et al., 2025; Ma et al., 2026; Xiao et al., 2026), or uses agreement among sampled paths as a proxy (Wang et al., 2023), and work on self-generated rewards raises concerns about signals the trained model can influence (Yuan et al., 2024; Lightman et al., 2023; Miao et al., 2023). Uncertainty is also used to decide whether to reflect (Lin et al., 2025; Su et al., 2025; Huang et al., 2026; Snell et al., 2025; Muennighoff et al., 2025) or to rescale GRPO updates (Xie et al., 2025; Zhao et al., 2026; Chen et al., 2025). In multi-turn objectives, however, calibrated confidence in the first answer is not evidence of capability: a deferred first answer is predictably wrong, so confidence is low exactly where the model could have answered. We therefore use a capability estimate fixed before training.

3 Why Reflection Rewards Pay for Deferral

Deferral in the observed behavior. A successful correction need not repair a genuine mistake: it can replace an answer that was only a placeholder. In the SFT model's rollouts on the MathVista training pool, nearly all successful corrections follow such placeholders, while genuinely attempted wrong answers are almost never fixed (Table 1). The SFT templates make this behavior available (Appendix B.2); we analyze when the reward favors it.

Reward and policy model. For an input x, the policy produces a first answer, a reflection, and a final answer, with correctness $(a_1, a_2) \in \{0,1\}^2$. The trajectory reward $R(a_1, a_2)$ also credits the first answer. Two differences govern the comparison:

$$h := R(0,1) - R(0,0), \qquad g := R(0,1) - R(1,1). \tag{1}$$

The *correction incentive* $h > 0$ rewards fixing an error over leaving it wrong. The *correction-incentive gap* g compares correction with preserving a correct first answer. SRPO's released implementation assigns $(R(0,1), R(1,1), R(1,0), R(0,0)) = (1, 0.75, 0.25, 0)$, so $g = 0.25$; we denote this reward variant by SRPO†. The reward specified in the SRPO paper (Wan et al., 2025) has $g = -0.25$.

The policy defers with probability d and otherwise attempts. An attempted first answer is correct with probability $s(x)$ and, if correct, is preserved. An incorrect attempt is fixed with probability $f(x)$. A deferred first answer is wrong, and its final answer is correct with probability $v(x)$. We call s the policy's *capability*: its accuracy conditional on attempting. Thus,

$$A_1 = (1-d)s, \qquad A_2 = (1-d)\big[s + (1-s)f\big] + dv. \tag{2}$$

Replacing an attempt with a deferral changes final accuracy by

$$\Delta(x) := v - s - (1-s)f. \tag{3}$$

An objective aligned solely with final accuracy should favor this change exactly when $\Delta > 0$. A correction bonus can shift that preference.

Table 1: **Deferral and attempted accuracy in the SFT model**. Groups use the number of correct first answers k in one half of each prompt's eight rollouts; statistics use the other half, with the roles swapped. Columns d, s, v are pooled conditional frequencies; Δ and vs are descriptive plug-in values. The genuine-fix rate satisfies $f \leq 0.004$ in every group. Counts and estimation details are in Appendix B.4.

k of 4	d	s	v	Δ	vs
0	0.87	0.21	0.26	$+0.04$	0.06
1	0.65	0.62	0.54	-0.08	0.33
2	0.46	0.80	0.64	-0.16	0.52
3–4	0.19	0.96	0.78	-0.18	0.75

Theorem 1 (Deferral bias)**.** *Locally, fix x, s, f, and v.*

(i) Expected reward. *Deferring instead of attempting changes expected reward by*

$$J_D - J_A = h\Delta + sg = h\big[\Delta + sg/h\big].$$

(ii) Idealized GRPO with $G = 2$. *Suppose $R(0,1) \neq R(1,1)$ and both exceed $R(0,0)$. Let $d = \sigma(\psi)$, where ψ controls only the probability of deferral. For two independent rollouts, the expected local on-policy score-function update with group-normalized advantages, no KL term, and no token weighting is proportional to*

$$d(1-d)\big[\Delta + \operatorname{sgn}(g)vs\big]$$

in the limit of a vanishing normalization stabilizer.

The normalized preference therefore consists of the final-accuracy term Δ plus a deferral bias*: sg/h in (i) and $\operatorname{sgn}(g)vs$ in (ii). For $g > 0$, this bias is nonnegative and scales with s, holding the other quantities fixed. Both biases vanish at $s = 0$.*

The expected-reward identity makes the mechanism explicit. An attempt produces outcome $(1,1)$ with probability s. Relative to a reward that values $(1,1)$ and $(0,1)$ equally, the gap adds sg to the preference for deferral. For example, with $s = v = 0.8$ and $f = 0$, attempting earns expected transition reward 0.60 under SRPO†, while deferring earns 0.80, despite identical final accuracy. At $G = 2$, normalization makes reward rankings decisive in the stated limit, so the gap enters through its sign. Proofs are in Appendix A; with damaged attempts or accidentally correct deferrals, an additional term enters the bias (Appendix A.2).

Corollary 1 (Consequences)**.** *Under the corresponding assumptions of Theorem 1, let β denote its deferral bias.*

(a) If $g > 0$, deferral is favored whenever $-\beta < \Delta < 0$, even though it lowers both accuracies.

(b) If deferral only postpones the same attempt ($v = s$, $f = 0$), then $g > 0$ makes full deferral expected-reward-maximizing for every $s > 0$: $A_1 = 0$ while $A_2 = s$ is unchanged.

(c) If $g < 0$, a positive preference for deferral requires $\Delta > 0$.

(d) Where $s = 0$, the sign of g has no effect on the deferral preference.

Descriptive evidence. Table 1 summarizes the SFT model using split-rollout cross-fitting (Chernozhukov et al., 2018): four rollouts assign each prompt to a success-count group, the other four estimate its group's statistics, and the roles are swapped. Groups with more observed successes have higher pooled attempted accuracy and lower deferral. Evaluating the $G{=}2$ bias of Theorem 1(ii) at these pooled rates gives vs from 0.06 to 0.75, exceeding $|\Delta|$ in each group, so under SRPO† the bias outweighs the accuracy cost of deferring. These are pooled, descriptive summaries of the mechanism, not per-prompt estimates of update directions.

Implications for reward design. Within this model, the analysis motivates three design criteria: (R1) make the gap non-positive on capable samples to remove the positive deferral bias; (R2) retain $h > 0$ so that correcting an error remains preferable to leaving it wrong; and (R3) allow more correction bonus where capability is low, since its associated bias is smaller.

A capability-dependent reward can meet these criteria by reducing the gap enough on capable samples while preserving $h > 0$. A uniform non-positive gap also removes the positive bias; Section 5.4 tests whether targeting protection offers an empirical advantage. Targeted protection, however, requires a capability proxy. This is difficult when first-answer success reflects both the ability to answer and the decision to attempt: $A_1 = (1-d)s$. We next examine which signals can distinguish these two factors.

4 METHOD: CAPABILITY-AWARE REWARDS

UASRL adds to the base reward a shaping term $\lambda C(c, a_1, a_2)$ driven by a per-sample signal c, chosen so that the resulting gap satisfies (R1)–(R3). Its name refers to uncertainty about whether the model will succeed on a sample, which we quantify by a fixed capability proxy; we denote the capability-dependent reward terms by r_{cal}.

4.1 WHICH SIGNALS CAN SERVE AS A CAPABILITY PROXY

Among the signals we analyze, only one fixed before training avoids both failure modes. With the shaping term, the gap becomes

$$g_{\text{eff}} = g - \lambda \big(\mathbb{E}[C \mid (1,1)] - \mathbb{E}[C \mid (0,1)]\big), \tag{4}$$

where the expectations are over the signal realized on rollouts with each outcome.

Proposition 1 (Signals under deferral)**.** *In the model of Theorem 1 with $f = 0$:*

(a) Confidence in the first answer. *If the policy reports c after choosing to attempt or defer, and $C = S(c, a_1)$ for a strictly proper scoring rule (Gneiting & Raftery, 2007) with equal maxima $S^\star$, the optimal reports are $c = 0$ after a deferral and $c = s$ after an attempt, giving $g_{\text{eff}} = g + \lambda[S^\star - S(s,1)] \geq g$. Under the Brier score (Brier, 1950) the increase is $\lambda(1-s)^2$.*

(b) Current success rate. *The policy's first-answer success rate converges to $(1-d)s$, which falls as deferral rises.*

(c) Success rate fixed before training. *A signal $c = c_e(x)$ held fixed gives $g_{\text{eff}}(x) = g - \Delta_C(c_e(x))$ with $\Delta_C(c) := \lambda[C(c,1,1) - C(c,0,1)]$, which training cannot change.*

Each part rules out or admits a natural signal. (a) Calibrated confidence rewards deferral: a deferred first answer is wrong by construction, so a calibrated report assigns it zero confidence and earns the maximal score (under the Brier score, the total expected preference for deferring rises by $\lambda s(1-s)$). (b) The current success rate is lowered by the very deferral it should detect, so protection weakens as deferral grows; restricting it to attempts or forcing attempts would avoid this at the cost of identifying deferrals or extra generation. (c) A fixed success rate breaks this loop. It measures $(1-d_0)s$ rather than s and so underestimates capability where the SFT model already defers, but in Table 1 groups with higher success rates also have higher attempt accuracy, which makes it a usable proxy at the group level.

4.2 CAPABILITY ESTIMATE

For each training prompt with answer $a^\star$, we draw N rollouts from the SFT model before RL and record the fraction of correct first answers:

$$c_e = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\big[a_{1,i} = a^\star\big]. \tag{5}$$

We use $N{=}8$ at temperature 0.7. The estimate is computed once, held fixed, and requires only the ground-truth answers already available for RL. On MathVista, 65% of training prompts have $c_e = 0$; a zero estimate does not mean zero capability, since the SFT model may defer or eight draws may miss a success.

4.3 REWARD

The reward makes a correct first answer worth more than a correction where c_e is high. We keep the base reward $R_{\text{base}} = r_{\text{acc}} + r_{\text{fmt}} + r_{\text{len}}$, where r_{acc} is SRPO† and r_{fmt}, r_{len} reward format and length, and add two capability terms and one auxiliary term:

$$R = R_{\text{base}} + \underbrace{r_{a1} + r_{\text{corr}}}_{r_{\text{cal}}} + r_{\text{grd}}, \tag{6}$$

$$r_{a1} = \begin{cases} 0.1\,(1 + c_e) & a_1 = 1 \\ -0.3\,c_e & a_1 = 0 \end{cases} \qquad r_{\text{corr}} = \begin{cases} 0.05 & (1,1) \\ 0.15\,(1 - c_e) & (0,1) \\ -0.3 & (1,0) \\ -0.1\,c_e & (0,0). \end{cases} \tag{7}$$

r_{a1} penalizes a wrong first answer in proportion to c_e, most where the model has shown it can answer and a wrong first answer is most likely a deferral. r_{corr} shrinks the correction bonus as c_e grows and penalizes damaging a correct answer. The auxiliary term r_{grd} is a lexical score $s_{\text{grd}} \in [-0.05, 0.13]$ for reflections that reference question content (Appendix B.3). It is awarded only when $a_1 = 1$, because every deferral is followed by an error-reporting reflection that an ungated score could reward.

Holding format and length equal, the gap and correction incentive are

$$g_{\text{eff}}(c_e) = 0.25 - 0.55\,c_e - s_{\text{grd}}, \qquad h(c_e) = 1.15 - 0.05\,c_e \tag{8}$$

(Appendix A.5). With $s_{\text{grd}} = 0$, the gap is negative for $c_e > 5/11$, that is, $c_e \geq 1/2$ on the $N{=}8$ grid, which meets (R1) where c_e is high; the grounding score moves this boundary between $c_e \geq 1/4$ and $c_e \geq 5/8$. $h > 0$ everywhere, which meets (R2). Below the boundary the gap stays positive where the SFT success rate is low; (R3) allows this to the extent that a low c_e reflects low capability rather than initial deferral. Protection is therefore not guaranteed on every capable prompt; the coefficients are empirical choices, varied in Appendix C.2. The estimate enters only the training reward; at test time the model generates exactly as the base method does.

5 EXPERIMENTS

Our experiments evaluate UASRL against reflection-trained baselines (Section 5.2), test whether deferral accounts for the differences between rewards (Section 5.3), assess the role of per-sample capability information (Section 5.4), isolate the contribution of each reward component (Section 5.5), and examine deferral beyond the templated initialization (Section 5.6).

5.1 SETUP

We train Qwen2.5-VL-7B (Bai et al., 2025) and InternVL3-8B (Zhu et al., 2025) on a math track (SRPO's released RL pool, evaluated on MathVista (Lu et al., 2024)) and a science track (ScienceQA (Lu et al., 2022)), with MMMU-Pro (Yue et al., 2024) held out. All methods start from the same templated SFT model: 68% of its training trajectories begin with a deferred first answer (the placeholder on MathVista, the option after the gold one on ScienceQA), followed by a templated reflection and the gold answer (Appendix B.2). All use GRPO with $G{=}2$, except the MathVista SRPO† run in Table 2 and the self-consistency variant ($G{=}4$); a matched $G{=}2$ SRPO† run is in Table 5. Baselines are SRPO† on both backbones and adaptations of SCoRe (Kumar et al., 2025) and VL-Rethinker (Wang et al., 2025) on Qwen. We report first- and final-answer accuracy A_1 and A_2 and the deferral rate d. Details are in Appendix B.

5.2 MAIN RESULTS

Table 2 compares first and final answers across two backbones and three benchmarks. Every reflection-trained baseline raises A_2 well above its own A_1, but its A_1 falls below Base in all six settings; on Qwen ScienceQA, SFT alone lowers A_1 from 77.4 to 37.4. UASRL has the highest A_1 among trained methods in every setting, exceeds Base in four of six, and has the highest A_2 among RL-trained methods, with the smallest gap between the two. **The final-answer gains of the baselines largely compensate for degraded first answers, whereas UASRL reaches its final accuracy mostly with the first answer.**

Table 2: **UASRL improves first answers across backbones and benchmarks.** First-answer (A_1) and final-answer (A_2) accuracy, in %. Subscripts: A_1 relative to Base; A_2 relative to the method's own A_1, measuring reliance on reflection. SCoRe and VL-Rethinker are re-implemented on the shared SFT checkpoint (Appendix B.5); MMMU-Pro uses four options and is held out. The Qwen MathVista SRPO† run uses G=4; the matched G=2 run is in Table 5. ↑: higher is better; bold: best among trained methods. red: lower,green: higher. Best view in color.

	MathVista (Lu et al., 2024)		MMMU-Pro (Yue et al., 2024)		ScienceQA (Lu et al., 2022)	
Method	A_1↑	A_2↑	A_1↑	A_2↑	A_1↑	A_2↑
Qwen2.5-VL-7B (Bai et al., 2025)						
Base	57.4	–	41.1	–	77.4	–
SFT	54.9(-2.5)	66.1(+11.2)	27.6(-13.5)	47.7(+20.1)	37.4(-40.0)	81.5(+44.1)
SCoRe (Kumar et al., 2025)	52.6(-4.8)	64.0(+11.4)	25.9(-15.2)	44.2(+18.3)	40.0(-37.4)	79.5(+39.5)
SRPO† (Wan et al., 2025)	49.9(-7.5)	67.3(+17.4)	21.2(-19.9)	47.0(+25.8)	10.3(-67.1)	82.9(+72.6)
VL-Rethinker (Wang et al., 2025)	42.3(-15.1)	64.5(+22.2)	23.1(-18.0)	47.2(+24.1)	33.0(-44.4)	83.4(+50.4)
UASRL (ours)	**61.3**(+3.9)	**67.8**(+6.5)	**36.4**(-4.7)	**49.1**(+12.7)	**80.2**(+2.8)	**84.2**(+4.0)
InternVL3-8B (Zhu et al., 2025)						
Base	71.0	–	38.2	–	93.4	–
SFT	68.5(-2.5)	76.4(+7.9)	26.8(-11.4)	51.7(+24.9)	52.0(-41.4)	92.0(+40.0)
SRPO† (Wan et al., 2025)	63.5(-7.5)	77.0(+13.5)	20.5(-17.7)	51.5(+31.0)	24.7(-68.7)	93.1(+68.4)
UASRL (ours)	**73.5**(+2.5)	**77.6**(+4.1)	**40.0**(+1.8)	**52.4**(+12.4)	**91.5**(-1.9)	**93.2**(+1.7)

Table 3: **SRPO† inverts ScienceQA answers.** Answer transitions on ScienceQA (Qwen2.5-VL-7B, 2,000 questions), in %. (a_1, a_2): correctness of the first and final answers. Changed: first and final options differ. Next: the first option cyclically follows the model's own final option. SRPO†'s $(1, 0)$ outcomes are accidental matches of deferral, not damaged attempts: in 202 of 205, the first option follows the final one.

Method	$(1,1)$	$(0,1)$	$(1,0)$	$(0,0)$	A_1↑	A_2↑	Changed	Next↓
SFT	30.8	50.6	6.6	12.0	37.4	81.5	61.4	60.6
SRPO†	0.0	82.9	10.3	6.9	10.3	82.9	100.0	97.0
UASRL (ours)	80.2	4.0	0.0	15.8	**80.2**	**84.2**	4.0	**2.7**

5.3 Deferral Separates the Rewards

If the rewards differed in how well the model answers, attempt accuracy would change; if they differed in deferral, the deferral rate would change while final accuracy stays fixed. On ScienceQA (Table 3), SRPO† changes every answer, and its first option cyclically follows its own final option on 97% of questions, including 96% of those with three or four options, where a uniform choice would give 50% or 33%; UASRL changes only 4%. On MathVista (Table 4), A_1 varies by 50 points across rewards while A_2 varies by less than one, and in every run most of the change in A_1 comes from deferral; no run fixes more than 0.4% of wrong attempts. UASRL's lower pooled attempt accuracy reflects the harder questions it now attempts: on the questions both SFT and UASRL attempt, their accuracy is equal (76.5% vs. 76.6%), and on those only UASRL attempts, its first answer is as accurate as SFT's answer after deferring (44.5% vs. 43.9%). The reduction concentrates where the SFT success rate is high and UASRL's gap is negative (Figure 2), although deferral also falls where the gap stays positive, which the gap's sign alone does not explain. **The rewards change whether the model attempts, not how well it answers**, consistent with the incentive of Theorem 1.

5.4 Signals and Forms of Protection

Section 4 argues that protection should be targeted by a capability estimate fixed before training. Table 5 replaces UASRL's signal while keeping everything else. Rewards that remove the bias uniformly improve on SRPO† but stay below UASRL. Shuffling c_e across prompts keeps deferral low but gives the lowest A_1 among protected rewards, so where protection lands matters. Confidence-based signals fail as Proposition 1 predicts: verbalized confidence increases deferral, and a fixed

Table 4: **Rewards differ mainly in deferral.** MathVista (Qwen2.5-VL-7B), in %. d: deferral rate; s: accuracy of attempted first answers; v: final accuracy after a deferral. ΔA_1: change from SFT, split by $A_1 = (1-d)s$ into a deferral and an attempt part (midpoint decomposition; an accounting, not a causal, split). *G=4 (for SRPO† also a KL penalty and one epoch). §Also rewards answer changes (Appendix B.5).

Method	$A_1\uparrow$	$A_2\uparrow$	$d\downarrow$	s	v	ΔA_1 from deferral	ΔA_1 from attempts
SFT	54.9	66.1	27.7	75.9	44.0	–	–
SRPO† *	49.9	67.3	35.0	76.8	49.7	−5.6	+0.6
Binary r_{cal}	59.3	67.5	17.8	72.1	46.1	+7.3	−2.9
Continuous r_{cal}	60.1	67.5	16.6	72.1	44.0	+8.2	−3.0
Ungated r_{grd} §	23.0	67.7	72.2	82.7	61.9	−35.3	+3.4
Self-consistency*	11.0	67.1	86.7	82.7	64.7	−46.8	+2.9
UASRL (ours)	**61.3**	**67.8**	**13.5**	70.9	48.1	+10.4	−4.0

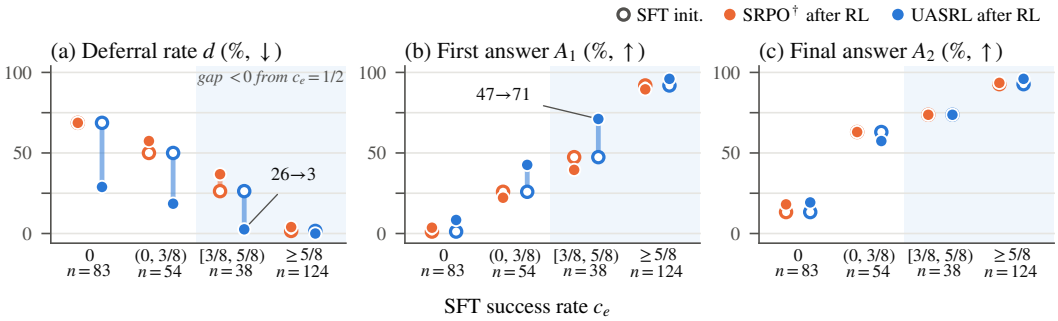


Figure 2: **UASRL nearly removes deferral where the SFT success rate is high.** MathVista (Qwen2.5-VL-7B) test questions grouped by the eight-rollout SFT success rate c_e. Each segment runs from the SFT initialization (hollow) to the model after RL (filled); SRPO† is the G=4 run of Table 2. Shaded: groups containing the prompts where UASRL's gap is negative. Exact values in Table 15.

answer likelihood, near-saturated and thus nearly uniform, leaves A_1 at the SFT level. **Per-sample capability information, fixed before training, distinguishes UASRL**; as these are single runs, we regard its margin over the strongest uniform control as preliminary.

5.5 Reward Components

Table 4 also isolates the reward components (component matrix in Table 13). Adding the capability terms lowers d from 32.5% under the matched SRPO† run to about 17%, and gated grounding lowers it to 13.5%. Ungated auxiliary rewards do the opposite: ungated grounding with an answer-change bonus and a self-consistency bonus raise d to 72% and 87% with A_2 unchanged. At initialization, the ungated run already pays more for deferral trajectories, entirely through its answer-change bonus rather than grounding (Table 14, Appendix C.3); the self-consistency bonus does not, so its collapse emerges during training. **Auxiliary rewards that deferral trajectories can earn amplify deferral unless gated on a correct first answer.** The binary variant keeps SRPO†'s outcome ranking yet defers far less than the matched SRPO† run (17.8% vs. 32.5%); no other term reorders the outcomes at initialization, so the two-rollout analysis does not explain this reduction. The gain holds across three GRPO seeds (61.0 ± 0.6 vs. 49.4 ± 1.6 for SRPO†) and reward coefficients, but requires more than N=2 rollouts for c_e (Appendix C.2).

5.6 Beyond the Templated Initialization

To test whether deferral is an artifact of the templates, we start RL from the base model with only the reflection prompt, which has seen no wrong-first examples and almost never defers (Table 6;

Table 5: **Per-sample capability targeting gives the highest A_1.** MathVista (Qwen2.5-VL-7B), in %; subscripts relative to SFT ($A_1/A_2 = 54.9/66.1$). All rows share UASRL's SFT initialization and GRPO configuration and differ only in the reward, except ‡, which starts from an SFT checkpoint that also emits a confidence tag ($A_1/A_2/d = 54.9/66.2/26.2$). Prediction: the deferral bias on capable prompts implied by Section 3. $^{\P}\lambda = 0.3$, selected on a validation split. §Fixed per-prompt likelihood, $c_m \in [0.67, 1.0]$. $^{\sharp}$At zero grounding.

	Method	Prediction	$A_1\uparrow$	$A_2\uparrow$	$d\downarrow$
No capability signal	SRPO† ($g = 0.25$)	bias + (Thm. 1)	52.4 (-2.5)	66.9 (+0.8)	32.5
	outcome-only ($g = 0$)	no bias	54.8 (-0.1)	67.2 (+1.1)	28.3
	SRPO† $+\lambda\mathbb{1}[a_1{=}1]^{\P}$	bias − iff $\lambda > 0.25$	56.3 (+1.4)	67.1 (+1.0)	26.4
	SRPO as specified ($g = -0.25$)	bias − everywhere	58.4 (+3.5)	67.2 (+1.1)	22.8
No per-sample information	UASRL reward, $c \equiv 1$	bias − everywhere	56.9 (+2.0)	65.3 (-0.8)	24.1
	UASRL reward, c_e shuffled	bias − on random prompts	53.5 (-1.4)	67.7 (+1.6)	18.4
Confidence-based signal	answer likelihood $c_m{}^{\S}$	bias − almost everywhere	54.5 (-0.4)	67.4 (+1.3)	26.0
	verbalized confidence, Brier‡	gap raised (Prop. 1a)	51.7 (-3.2)	64.5 (-1.6)	30.2
	UASRL (ours), c_e	bias − where $c_e \geq 1/2^{\sharp}$	**61.3** (+6.4)	**67.8** (+1.7)	**13.5**

Table 6: **Deferral emerges without templates.** MathVista (Qwen2.5-VL-7B), starting from the base model with a reflection prompt and no SFT (Appendix B.6), in %; subscripts relative to the initialization. d: deferral rate; genuine correction: share of wrong attempted first answers fixed in the final answer.

Method	$A_1\uparrow$	$A_2\uparrow$	$d\downarrow$	Genuine correction$\uparrow$
Initialization	57.3	57.6	0.3	0.9
SRPO†	46.6 (-10.7)	63.1 (+5.5)	17.6	3.6
Uniform protection	58.2 (+0.9)	62.7 (+5.1)	2.9	4.0
UASRL (ours)	**61.9** (+4.6)	**64.8** (+7.2)	**2.4**	**5.1**

Appendix B.6). SRPO† raises deferral to 17.6% and lowers A_1 by 10.7 points while its A_2 rises, so final accuracy again hides the failure; UASRL keeps deferral near its initial level and gives the highest A_1 and A_2. **Deferral can emerge from the reward alone**; as this is a single setting, we treat it as preliminary evidence.

6 Conclusion & Limitations

Conclusion. A model trained to answer twice can learn to defer: to withhold its first answer and give it in the second attempt. We showed that when a trajectory-level reward values a correction above a preserved correct answer, it creates a deferral bias that grows with the model's capability, and that deferral which merely postpones the answer leaves final accuracy unchanged while the first answer degrades. In a controlled setting, deferral accounts for nearly all corrections and for most of the differences in first-answer accuracy across rewards. UASRL counters this bias with a capability estimate fixed before RL, improving first-answer accuracy across two backbones and three benchmarks while keeping final accuracy competitive. More broadly, reflection methods should report first-answer accuracy and answer transitions alongside final accuracy, since final accuracy alone cannot distinguish correction from deferral.

Limitations. We study deferral mainly in a controlled setting whose SFT templates make it available from the start, which allows it to be measured directly; our non-templated experiment indicates that it also arises without templates, and extending the analysis to initializations with substantial genuine correction is a natural next step. Our analysis is local: it characterizes the incentive a reward creates rather than the full training dynamics of a shared-parameter policy. The capability estimate is a proxy computed from a finite number of SFT rollouts. More direct estimates of conditional capability, broader comparisons between targeted and uniform protection, and rewards for multiple revision steps are promising directions.

AI USE STATEMENT

AI Use Statement. Generative AI tools were used to assist with manuscript preparation, primarily for English language editing and improving clarity and readability, as well as for limited assistance with code development and refinement. All research ideas, methodological design, experimental settings, analyses, and conclusions were developed, reviewed, and validated by the authors. The authors take full responsibility for the content of the manuscript.

REFERENCES

Shuai Bai et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.

Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mozolevskyi, Muayad Ali, and Waseem AlShikh. Reflect, retry, reward: Self-improving llms via reinforcement learning. *arXiv preprint arXiv:2505.24726*, 2025.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Yi Ding and Ruqi Zhang. Sherlock: Self-correcting reasoning in vision-language models. *arXiv preprint arXiv:2505.22651*, 2025.

Yi Ding, Ziliang Qiu, Bolian Li, and Ruqi Zhang. Learning self-correction in vision-language models via rollout augmentation. *arXiv preprint arXiv:2602.08503*, 2026.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2023.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Daya Guo et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31176–31184, 2026.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhizheng Jiang, Kang Zhao, Weikai Xu, Xinkui Lin, Wei Liu, Jian Luan, Shuo Shang, and Peng Han. R^3: Replay, reflection, and ranking rewards for LLM reinforcement learning. *arXiv preprint arXiv:2601.19620*, 2026.

Saurav Kadavath et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Aviral Kumar et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2025.

Yibo Li et al. ConfTuner: Training large language models to express their confidence verbally. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Chenyu Lin et al. Learning to think fast and slow for visual language models. *arXiv preprint arXiv:2511.16670*, 2025.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Pan Lu et al. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2024.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S^2R: Teaching LLMs to self-verify and self-correct via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 22632–22654, 2025.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards. *arXiv preprint arXiv:2603.09117*, 2026.

Aman Madaan et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2023.

Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*, 2023.

Niklas Muennighoff et al. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems*, 2024.

Zhihong Shao et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Taiwei Shi, Sihao Chen, Bowen Jiang, Linxin Song, Longqi Yang, and Jieyu Zhao. Experiential reinforcement learning. *arXiv preprint arXiv:2602.13949*, 2026.

Noah Shinn, Federico Cassano, et al. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, 2022.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2025.

Alexander Stangel, David Bani-Harouni, et al. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *International Conference on Learning Representations (ICLR)*, 2026.

DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qinqing Zheng. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *International Conference on Learning Representations (ICLR)*, 2025.

Katherine Tian et al. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *Conference on Empirical Methods in Natural Language Processing*, 2023.

Teun van der Weij, Felix Hofstätter, Ollie Jaffe, Samuel F. Brown, and Francis Rhys Ward. AI sandbagging: Language models can strategically underperform on evaluations. *arXiv preprint arXiv:2406.07358*, 2024.

Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, et al. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement learning. *arXiv preprint arXiv:2506.01713*, 2025.

Hanbin Wang et al. Teaching large reasoning models effective reflection. *arXiv preprint arXiv:2601.12720*, 2026.

Haozhe Wang et al. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025.

Xuezhi Wang et al. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations*, 2023.

Zhijie Wang. GRPO and reflection reward for mathematical reasoning in large language models. *Applied and Computational Engineering*, 154:161–166, 2025.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via fine-grained reward structure and credit assignment. *arXiv preprint arXiv:2505.11821*, 2025.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *International Conference on Learning Representations*, 2023.

Mingyuan Wu, Jingcheng Yang, Shengyi Qian, Xudong Wang, Jize Jiang, Qifan Wang, Aashu Singh, Khoi Pham, Fei Liu, Zhaolun Su, Zhuokai Zhao, Klara Nahrstedt, Jianyu Wang, and Hanchao Yu. SVR-R1: Bootstrapping multi-modal reasoning with self-verification in reinforcement learning. *arXiv preprint arXiv:2607.10966*, 2026.

Wenyi Xiao, Xinchi Xu, and Leilei Gan. Vl-calibration: Decoupled confidence calibration for large vision-language models reasoning. *arXiv preprint arXiv:2604.09529*, 2026.

Can Xie et al. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*, 2025.

Miao Xiong et al. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *International Conference on Learning Representations*, 2024.

Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Xiang Yue et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7113–7139, 2024a.

Renrui Zhang et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*, 2024b.

Qiannian Zhao et al. Know what you know: Metacognitive entropy calibration for verifiable RL reasoning. *arXiv preprint arXiv:2602.22751*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning*, 2024.

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

Xudong Zhu, Jiachen Jiang, Mohammad Mahdi Khalili, and Zhihui Zhu. From emergence to control: Probing and modulating self-reflection in language models. *Transactions on Machine Learning Research*, 2026.

A PROOFS

Throughout, x is fixed and s, f, v are held fixed; the results are local, concerning the expected reward and the direction of the expected update at the current policy, not the full training trajectory, along which these quantities change. A correct attempted first answer is kept. Outcomes and their probabilities are

	$(1,1)$	$(0,1)$	$(0,0)$
Attempt	s	$(1-s)f$	$(1-s)(1-f)$
Defer	0	v	$1-v$

Note that $R(1,1) - R(0,0) = h - g$.

A.1 PROOF OF THEOREM 1

(i) Subtracting $R(0,0)$ from every reward, $J_A = s\,(h-g) + (1-s)f\,h$ and $J_D = v\,h$. Hence $J_D - J_A = h\,[\,v - (1-s)f\,] - s\,(h-g) = h\,[\,v - s - (1-s)f\,] + s\,g = h\,[\,\Delta + s\,g/h\,]$. □

(ii) Let $d = \sigma(\psi)$, so the score of ψ is $1-d$ on a deferred rollout and $-d$ on an attempt. With $G = 2$, a group with two rollouts of the same kind contributes nothing: both scores are equal and the advantages sum to zero. A mixed group occurs with probability $2d(1-d)$. If its rewards differ, the group-normalized advantages are $\pm a$, where $a > 0$ becomes independent of the reward values in the limit of a vanishing stabilizer (see *Normalization* below), and the contribution to the update is $\frac{1}{2}[A_D(1-d) + A_A(-d)] = \frac{1}{2}A_D$. The expected update is therefore $a\,d(1-d)\,[P(D \succ A) - P(A \succ D)]$, where $D \succ A$ means the deferred rollout has the larger reward.

If $g > 0$, the ranking is $(0,1) \succ (1,1) \succ (0,0)$, and

$$P(D \succ A) = v\,[\,s + (1-s)(1-f)\,], \qquad P(A \succ D) = (1-v)\,[\,s + (1-s)f\,],$$
$$P(D \succ A) - P(A \succ D) = v(1+s) - s - (1-s)f = \Delta + vs.$$

If $g < 0$, the ranking is $(1,1) \succ (0,1) \succ (0,0)$, and

$$P(D \succ A) = v\,(1-s)(1-f), \qquad P(A \succ D) = s + (1-s)\,f\,(1-v),$$
$$P(D \succ A) - P(A \succ D) = v(1-s) - s - (1-s)f = \Delta - vs.$$

Ties, $(0,1)$ against $(0,1)$ and $(0,0)$ against $(0,0)$, have zero advantage. □

Normalization. Group normalization and its biases are discussed by Liu et al. (2025). With two distinct rewards $r_a \neq r_b$ and advantages $(r_i - \bar{r})/(\sigma + \epsilon)$, the magnitude is $a = \frac{|r_a - r_b|/2}{|r_a - r_b|/2 + \epsilon}$ for the population standard deviation (and the same up to a constant factor for the sample standard deviation), which is constant only in the limit $\epsilon/|r_a - r_b| \to 0$. Part (ii) is exact in that limit; with a finite ϵ (our trainer uses 10^{-4}) it is an approximation whose quality depends on how close the rewards of the compared rollouts are, and reward differences between outcomes can be small (e.g., 0.025 for UASRL at $c_e = 1/2$, and smaller with the continuous length and grounding terms). Within the fixed-outcome model, two rollouts with the same outcome have equal reward and cancel; with auxiliary terms not determined solely by the correctness outcome, an attempt and a deferral with the same outcome can receive different rewards and contribute to the update. Clipping, per-token weighting, and shared parameters are outside the abstraction.

Larger groups. For $G > 2$ the advantages depend on reward magnitudes. As $G \to \infty$ the group mean and standard deviation concentrate, and the expected update of ψ becomes proportional to $d(1-d)(J_D - J_A)/\sigma_R$, whose sign is given by (i).

A.2 DAMAGED ATTEMPTS AND ACCIDENTAL MATCHES

Let a correct attempt be kept with probability q (so it becomes $(1,0)$ with probability $1-q$), and let a deferral produce a correct first answer by accident with probability u, in which case the final answer is wrong, as for the inverted answers of Section 5.3; $u + v \leq 1$, and $A_1 = (1-d)\,s + d\,u$. The outcome probabilities become

	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$
Attempt	sq	$s(1-q)$	$(1-s)f$	$(1-s)(1-f)$
Defer	0	u	v	$1-u-v$

With $\Delta := v - sq - (1-s)f$ and $r := R(1,0) - R(0,0)$, subtracting $R(0,0)$ from every reward gives $J_A = sq\,(h-g) + s(1-q)\,r + (1-s)f\,h$ and $J_D = u\,r + v\,h$, so

$$J_D - J_A = h\,\Delta + s\,q\,g + \big(u - s(1-q)\big)\,r.$$

The capability-proportional term becomes sqg; accidental matches add $u\,r$ to the reward of deferring, and damaged attempts subtract $s(1-q)\,r$. For SRPO†, $r = 0.25$; for UASRL, $r = 0.05 + 0.5c_e + s_{\text{grd}}$. Theorem 1(i) is the case $q = 1$, $u = 0$.

The full bias is $sqg + (u - s(1-q))\,r$, and a negative gap no longer guarantees that it is non-positive. For example, UASRL's reward at $c_e = 1/2$ with zero grounding has $(R(1,1), R(0,1), R(1,0), R(0,0)) = (0.95, 0.925, 0.10, -0.20)$ and $g = -0.025$; with $s = 0.5$, $q = 1$, $f = 0$, $u = 0.4$, and $v = 0.45$, deferring lowers both A_1 (from 0.5 to 0.4) and A_2 (from 0.5 to 0.45) yet raises the expected reward from 0.375 to 0.426. Removing the bias in this model requires the full expression to be non-positive. This extension concerns the expected reward only; we do not derive a corresponding G=2 result.

A.3 Proof of Corollary 1

(a) With $g > 0$ the bias β is nonnegative, and positive when $s > 0$ (and $v > 0$ in the G=2 case), so the net preference $\Delta + \beta$ is positive whenever $-\beta < \Delta$; if also $\Delta < 0$, deferral lowers A_2, and it always lowers $A_1 = (1-d)s$ when $s > 0$. Both biases are zero at $s = 0$ and increasing in s (for fixed v in the G=2 case). (b) With $v = s$ and $f = 0$, $\Delta = 0$ and the net preference equals $\beta > 0$ for every $s > 0$. By (i), J is linear in d with positive slope, so it is maximized at $d = 1$, giving $A_1 = 0$ and $A_2 = v = s$, the same final accuracy as $d = 0$. (c) With $g < 0$, $\beta \le 0$, so a positive net preference requires $\Delta > 0$. (d) $\beta = 0$ when $s = 0$. □

A.4 Proof of Proposition 1

(a) With $f = 0$, outcome $(0,1)$ arises only from deferrals and outcome $(1,1)$ only from attempts. The policy reports c knowing only whether it deferred and its capability s. After a deferral, $a_1 = 0$ with certainty, so a proper score is maximized by $c = 0$, giving $S(0,0) = S^\star$. After an attempt, $a_1 = 1$ with probability s, so the expected score is maximized by $c = s$, and on $(1,1)$ rollouts the realized score is $S(s,1)$. Hence $\mathbb{E}[C \mid (1,1)] - \mathbb{E}[C \mid (0,1)] = S(s,1) - S^\star \le 0$, and $g_{\text{eff}} = g + \lambda\,[S^\star - S(s,1)]$. For a strictly proper rule $S(s,1) < S^\star$ unless $s = 1$. For the Brier score, $S(c,a) = -(c-a)^2$, the increase is $\lambda(1-s)^2$. This is the conditional gap. The expected score is 0 after a deferral and $-s(1-s)$ after an attempt, so the total expected preference for deferring rises by $\lambda\,s(1-s)$.

(b) The first-answer success rate of the current policy is $A_1(x) = (1-d)s$ by Eq. 2, decreasing in d.

(c) If $c = c_e(x)$ does not depend on θ, it takes the same value on every rollout of x, so $\mathbb{E}[C \mid (1,1)] - \mathbb{E}[C \mid (0,1)] = C(c_e,1,1) - C(c_e,0,1)$, and $g_{\text{eff}}(x) = g - \Delta_C(c_e(x))$. □

A.5 Full Reward and Pairwise Gap

Table 7 lists the reward for all four outcomes, excluding the format and length terms, which the pairwise statements assume equal across the compared trajectories.

Gap and correction incentive. Subtracting the $(1,1)$ row from the $(0,1)$ row gives $R(0,1) - R(1,1) = 0.25 + (-0.1 - 0.4c) + (0.1 - 0.15c) - s_{\text{grd}} = 0.25 - 0.55c - s_{\text{grd}}$. The r_{a1} contribution is negative for all c; the r_{corr} contribution $0.1 - 0.15c$ is positive for $c < 2/3$. Subtracting the $(0,0)$ row from the $(0,1)$ row gives the correction incentive $h = R(0,1) - R(0,0) = 1.15 - 0.05c \in [1.10, 1.15]$, positive for every c and independent of s_{grd}; the base reward has $h = 1$.

Table 7: Reward by outcome, excluding format and length terms. $c = c_e$; s_{grd} is the grounding score, earned only when $a_1 = 1$.

(a_1, a_2)	r_{acc}	r_{a1}	r_{corr}	r_{grd}	Total
$(1,1)$	0.75	$0.1(1{+}c)$	0.05	s_{grd}	$0.90 + 0.1c + s_{\text{grd}}$
$(0,1)$	1.0	$-0.3c$	$0.15(1{-}c)$	0	$1.15 - 0.45c$
$(1,0)$	0.25	$0.1(1{+}c)$	-0.3	s_{grd}	$0.05 + 0.1c + s_{\text{grd}}$
$(0,0)$	0	$-0.3c$	$-0.1c$	0	$-0.4c$

Boundary. With $s_{\text{grd}} = 0$ the gap is negative iff $c > 5/11$, which on the grid $\{0, 1/8, \ldots, 1\}$ is $c \geq 1/2$ ($c = 3/8$ gives $+0.044$; $c = 1/2$ gives -0.025). The grounding score lies in $[-0.05, 0.13]$: the smallest grid value with a negative gap is $c = 1/4$ at $s_{\text{grd}} = 0.13$ (-0.0175) and $c = 5/8$ at $s_{\text{grd}} = -0.05$ (-0.044). For another rollout count N, the protected set is recomputed from $0.25 - 0.55c - s_{\text{grd}} < 0$.

B Implementation Details

B.1 Training

Table 8: Training hyperparameters (Qwen2.5-VL-7B; LoRA in bf16, visual encoder frozen). The math track trains on 5,000 prompts from SRPO's released RL pool (Wan et al., 2025), the science track on 5,000 image-bearing ScienceQA training questions. Two runs differ from the GRPO column: the MathVista SRPO† checkpoint (G=4, β_{KL}=0.04, one epoch) and the self-consistency variant (G=4).

	SFT	GRPO
LoRA rank / α / dropout	64 / 128 / 0.05	64 / 128 / 0.05
LoRA targets	all linear layers	`q,k,v,o_proj`
Learning rate (AdamW)	$2{\times}10^{-5}$	10^{-6}
Batch $\times$ grad. accumulation	1×16	1×8
Rollouts per prompt G / KL β	–	2 / 0
Loss	–	token-level (DAPO)
Max. length	1,024	1,024 (completion)

B.2 SFT Data

We use *templated* reflection trajectories in the format `<think> <answer> <reflection> <think> <answer>`, to isolate the effect of the RL reward from SFT data quality. For each prompt the base model generates a response. If it is correct (32%), the trajectory confirms it; otherwise (68%), the first answer is a deferral, a templated reflection reports an error, and the second answer gives the gold answer. On the math track the deferred first answer is the placeholder `\boxed{wrong}`; on ScienceQA it is the option cyclically following the gold option. These templates make deferral available from the start. All methods use the same system prompt:

> *Solve the user's question step by step. First, think about the reasoning process internally and write it inside `<think>` and `</think>` tags. Then provide the first answer in LaTeX format, wrapped with `$...$`, and the final result must use `\boxed{}`. Wrap this answer inside `<answer>` and `</answer>` tags. After that, perform a critical self-reflection on the previous reasoning and answer, writing the reflection inside `<reflection>` and `</reflection>` tags. Then, based on the reflection, generate a new reasoning process and a new answer.*

B.3 Capability Estimate and Reward

c_e is the fraction of correct first answers among N=8 SFT rollouts (temperature 0.7, top-p 0.95), computed once per training prompt and shared by all variants; on the math pool its mean is 0.16

(65% zeros), on ScienceQA 0.44. The reward is $r_{\text{acc}} + r_{\text{fmt}} + r_{\text{len}} + r_{a1} + r_{\text{corr}} + r_{\text{grd}}$: r_{acc} is SRPO†'s transition reward; $r_{\text{fmt}} = 0.5$ for the exact five-block format; r_{len} (at most 0.1) favors completions about twice the length of the first answer; r_{a1} and r_{corr} follow Eq. 7. Answers are scored with `math_verify`. Edge cases outside the four-outcome model (fewer than two answer blocks, or an unparseable gold answer) occur in 0.17% of SFT rollouts.

Grounding score. $r_{\text{grd}} \in [-0.05, 0.13]$ is a deterministic lexical score, awarded only when $a_1 = 1$: $+0.05$ if the reflection repeats a number from the question, $+0.04$ each for visual words (e.g., *figure*, *chart*) and content words (e.g., *equation*, *angle*), and -0.05 for two or more template phrases (e.g., *I notice an error*) when the score is otherwise below 0.05. It is a proxy for engagement with the question, not a validated measure of grounding.

B.4 Measuring Deferral

On MathVista, a first answer containing the placeholder is a deferral; all others are attempts. For Table 1, each prompt's eight rollouts are split into halves: the success count k in one half assigns the group and the other half estimates d, s, v, f, and vice versa; counts are pooled within groups (Table 9). The same rule on greedy test outputs gives d elsewhere. On ScienceQA, a deferral is a first option that cyclically follows the model's own final option; for two-option questions any change has this form.

Table 9: Counts behind Table 1.

k of 4	Halves	Attempts	Deferrals
0	7,171	3,831	24,853
1	1,031	1,454	2,670
2	639	1,380	1,176
3–4	1,159	3,749	887

B.5 Baselines and Controls

All start from the same SFT checkpoint, except the verbalized-confidence control (Table 5).

- **SRPO†**: $r_{\text{acc}} + r_{\text{fmt}} + r_{\text{len}}$ ($g = 0.25$).
- **Binary** r_{cal}: a two-level term split at $c_e = 1/2$, adding $(+0.2, +0.1, -0.3, -0.1)$ to $(1,1), (0,1), (1,0), (0,0)$ above the split and $(+0.1, +0.3, -0.2, -0.1)$ below; it keeps SRPO†'s outcome ranking on every prompt. The ungated-grounding and self-consistency variants use this term.
- **Ungated grounding**: the binary term plus r_{grd} on every rollout and an answer-change bonus: $+0.15(1 - J)$ when the final answer differs from the first, where J is the token-level Jaccard similarity of the two reasoning passages, and -0.05 when the answer is unchanged and $J > 0.8$.
- **Self-consistency** (G=4): the binary term plus $+0.15$ when a rollout's final answer matches its group's majority and -0.05 otherwise, replacing r_{grd}.
- **Controls**: outcome-only ($R = \mathbb{1}[a_2 = 1]$); SRPO† $+\lambda\mathbb{1}[a_1 = 1]$ with $\lambda = 0.3$; the reward specified by SRPO ($g = -0.25$); and UASRL's reward with $c \equiv 1$ or with c_e permuted across prompts.
- **SCoRe** (Kumar et al., 2025): we follow its two-stage procedure and reward shaping. Stage I trains the second attempt while a KL penalty keeps the first attempt close to the initialization; Stage II trains both attempts with the shaping bonus $\alpha(r_2 - r_1)$, $\alpha = 10$, crediting each turn with its own reward. Unlike the original, it starts from our shared SFT checkpoint rather than the base model.
- **VL-Rethinker** (Wang et al., 2025): we follow its GRPO training with selective sample replay and forced rethinking, starting from the shared SFT checkpoint.

B.6 Non-Templated Initialization

This experiment (Table 6) uses Qwen2.5-VL-7B on the MathVista track, with the same training prompts, system prompt, GRPO configuration (Table 8), and evaluation protocol as the main experiments. The initialization is the base model with the reflection system prompt and no SFT. It contains no wrong-first examples, and its first answer is almost always an attempt: its A_1 (57.3) is close to that of the base model (57.4), and its final answer rarely differs from the first ($A_2 = 57.6$). Without a placeholder, a deferral cannot be read off the first answer alone, so we count a first answer as a deferral if it is wrong, differs from the final answer, and the same checkpoint answers the question correctly in a single pass without reflection. Genuine correction is the share of the remaining wrong first answers whose final answer is correct. Uniform protection uses UASRL's reward with $c \equiv 1$.

C Additional Results

C.1 Quality of the Capability Signal

Table 10 compares the SFT success rate c_e with the answer-likelihood confidence $c_m = \exp(-\overline{\mathrm{NLL}}_{A_1})$, the exponentiated mean token log-likelihood of the SFT model's greedy first answer. The success rate discriminates better and is better calibrated on every benchmark. c_m is nearly saturated: on ScienceQA it assigns scores above 0.999 to almost every sample, and its AUC (0.469) is below chance. Because both signals come from one fixed checkpoint, this difference reflects discrimination, not co-optimization. Figures 3 and 4 show per-sample distributions.

Table 10: **Signal quality across benchmarks.** We compare answer-likelihood confidence c_m and the SFT success rate c_e on discrimination (AUC, Spearman ρ), calibration (ECE, Brier), and practical utility (coverage and precision of the protected set $c_e \geq 1/2$, the zero-grounding boundary of Eq. 8 on the N=8 grid). Both signals come from the SFT checkpoint. The target is the correctness of the SFT model's greedy first answer; c_e uses 8 separately sampled rollouts (temperature 0.7), so target and estimate are independent draws. n: evaluated examples with a parseable answer, from subsets of 300 (MathVista testmini) and 200 (ScienceQA, MMMU-Pro) test questions.

		Discrimination				Calibration				Protection	
		AUC ↑		Spearman ρ ↑		ECE ↓		Brier ↓		Coverage	Precision
Benchmark	n	c_m	c_e	c_m	c_e	c_m	c_e	c_m	c_e	$c_e \geq \frac{1}{2}$	$c_e \geq \frac{1}{2}$
MathVista	299	0.791	**0.947**	0.503	**0.786**	0.473	**0.038**	0.459	**0.093**	47.5%	88.0%
ScienceQA	200	0.469	**0.776**	−0.052	**0.466**	0.635	**0.078**	0.635	**0.184**	48.5%	56.7%
MMMU-Pro	199	0.594	**0.854**	0.153	**0.608**	0.631	**0.089**	0.613	**0.133**	24.6%	83.7%

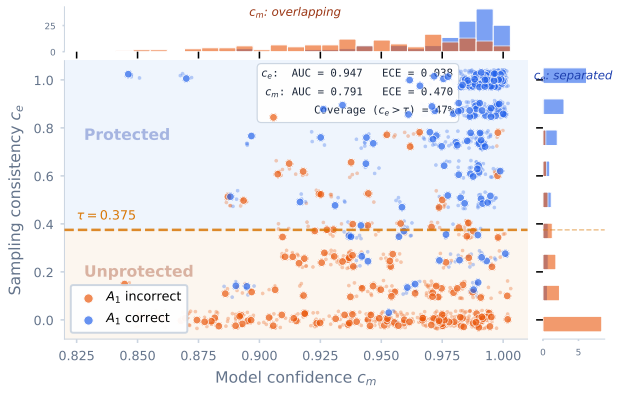
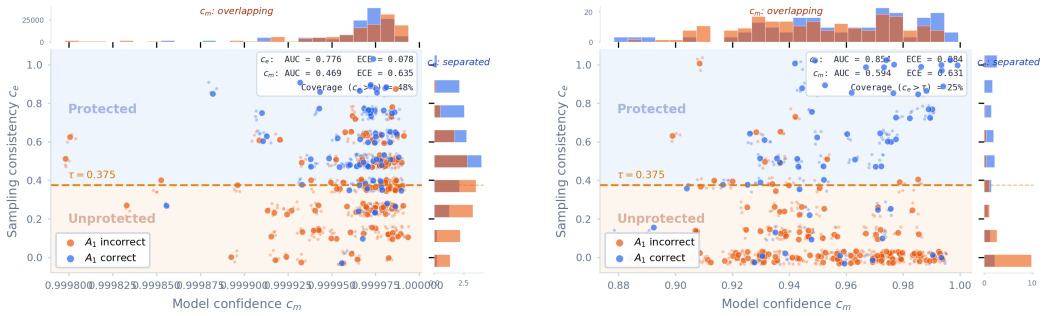


Figure 3: Per-sample c_m and c_e on MathVista for correct and incorrect first answers. c_m clusters near 1 for both; c_e separates them.



(a) ScienceQA (b) MMMU-Pro

Figure 4: Per-sample confidence on ScienceQA and MMMU-Pro, in the format of Figure 3.

C.2 Robustness

Seeds. We retrain SRPO† and UASRL with three GRPO seeds on MathVista, holding the SFT initialization and c_e fixed (Table 11). The A_1 gap (61.0 vs. 49.4) is far larger than the spread across seeds, and the standard deviation of UASRL's A_1 is about 2.7× smaller than SRPO†'s.

Table 11: Three GRPO seeds on MathVista (mean ± standard deviation), in %.

Method	$A_1 \uparrow$	$A_2 \uparrow$
SRPO†	49.4 ± 1.6	67.0 ± 0.5
UASRL (ours)	$\mathbf{61.0 \pm 0.6}$	$\mathbf{67.6 \pm 0.4}$

Coefficients and rollout budget. Table 12 varies the penalty-to-reward ratio of r_{a1}, the overall scale of r_{cal}, and the number of rollouts N used for c_e. A_1 stays above SFT in every setting except N=2, and A_2 stays between 66.5 and 67.9; with two rollouts c_e takes only three values and the protected set is coarse. The default coefficients are empirical choices, not values prescribed by the theory.

Table 12: **Sensitivity on MathVista** (Qwen2.5-VL-7B). Ratio: penalty-to-reward ratio α_-/α_+ of r_{a1}; scale: multiplier on r_{cal}; N: rollouts used to compute c_e. Default in bold. Changing the coefficients or N changes the protected set of Eq. 8.

Factor	Setting	A_1	A_2
Ratio	1 (symmetric)	56.5	67.4
	2	59.7	67.6
	3	**61.3**	**67.8**
	5	60.8	67.5
Scale	0.5×	58.9	67.5
	2×	59.4	67.0
N	2	54.0	66.5
	4	58.7	67.4
	8	**61.3**	**67.8**
	16	61.6	67.9

C.3 Reward Component Matrix

Table 13 lists which components each variant of Table 4 uses.

Table 14 applies each run's reward to the SFT rollouts at initialization (Section 5.5).

Table 13: **Reward components** (MathVista, Qwen2.5-VL-7B). All variants share the base model and SFT initialization; only the reward differs (the SRPO† and self-consistency runs use G=4; see Section 5.1). Binary: two reward levels split at $c_e = 1/2$ (Appendix B.5). d: deferral rate. Δ_{A_1}: change relative to SFT (A_1=54.9).

	Reward components			Results			
Variant	r_{cal}	r_{grd}	Gated	A_1	A_2	d	Δ_{A_1}
SRPO†	✗	✗	–	49.9	67.3	35.0	−5.0
Grounding only	✗	✓	yes	52.0	67.4	33.0	−2.9
Binary r_{cal}	binary	✗	–	59.3	67.5	17.8	+4.4
Continuous r_{cal}	cont.	✗	–	60.1	67.5	16.6	+5.2
Ungated r_{grd}†	binary	✓	no	23.0	67.7	72.2	−31.9
Self-consistency*	binary	✗	no	11.0	67.1	86.7	−43.9
UASRL (full)	**cont.**	✓	**yes**	**61.3**	**67.8**	**13.5**	**+6.4**

†Also rewards answer changes (Appendix B.5). *Replaces r_{grd} with an ungated self-consistency reward (G=4).

Table 14: **Realized reward differences at initialization.** Each run's reward applied term by term to the 40,000 SFT rollouts on the MathVista-track training prompts; mean on $(0, 1)$ (99.9% deferrals) minus mean on $(1, 1)$, pooled over prompts. Shared terms are identical across runs: r_{acc} +0.25, r_{fmt} 0, r_{len} −0.013; parser edge cases are 0.17% of rollouts. Flipped: share of within-prompt pairs with different outcomes whose total-reward ranking differs from that of $r_{\text{acc}} + r_{\text{cal}}$.

Run	r_{cal}	Auxiliary	Flipped (%)
Binary r_{cal}	+0.104	–	0.0
Ungated r_{grd}	+0.104	+0.142	0.0
Self-consistency	+0.104	−0.021	0.3
c_m (fixed)	−0.532	−0.040	0.0
UASRL	−0.123	−0.040	3.3

C.4 Transition Decomposition

Table 15 gives the values of Figure 2.

Figure 5 and Table 16 report answer transitions on 300 MathVista test examples, grouped by the SFT success rate c_e. UASRL's gain in initial accuracy is largest in the intermediate group; the group boundaries differ from those of Section 3. In the low group, UASRL answers correctly more often on the first attempt (27.2% vs. 13.9%) and corrects less often (10.8% vs. 26.6%), with similar final accuracy. Since all corrections on MathVista come from deferrals (Table 18), fewer 0→1 transitions here mean fewer deferrals, not weaker correction.

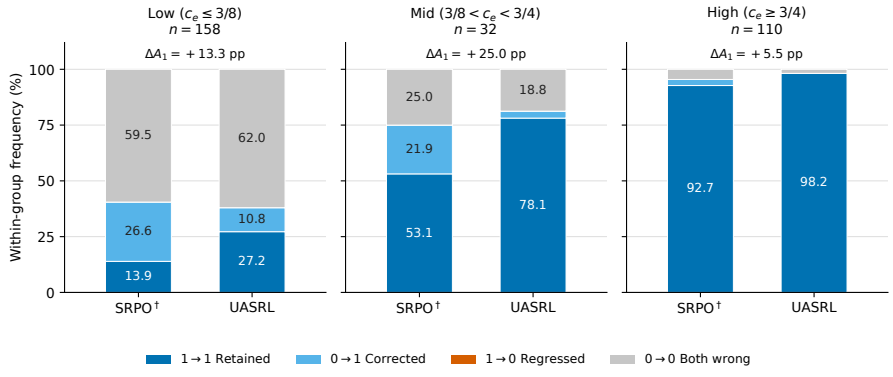


Figure 5: Answer transitions on 300 MathVista test examples, grouped by c_e. Bars sum to 100% within each group.

Table 15: **Deferral by SFT success rate.** MathVista test questions with an eight-rollout SFT success rate c_e (299 of the 1,000), grouped by c_e; the groups differ from the four-rollout groups of Table 1. d: deferral rate (%); A_1, A_2 in %.

	$c_e = 0$ (n=83)			$0 < c_e < 3/8$ (n=54)			$3/8 \le c_e < 5/8$ (n=38)			$c_e \ge 5/8$ (n=124)		
Method	d	A_1	A_2	d	A_1	A_2	d	A_1	A_2	d	A_1	A_2
SFT	69	1	13	50	26	63	26	47	74	2	92	93
SRPO†	69	4	18	57	22	63	37	39	74	4	90	94
UASRL	29	8	19	19	43	57	3	71	74	0	96	96

Table 16: Transition frequencies (%) on 300 MathVista test examples, grouped by c_e (Low: $c_e \le 3/8$, n=158; Mid: $3/8 < c_e < 3/4$, n=32; High: $c_e \ge 3/4$, n=110). Rows sum to 100.

Group	Method	1→1	0→1	1→0	0→0	ΔA_1
Low	SRPO†	13.9	26.6	0.0	59.5	+13.3
	UASRL	27.2	10.8	0.0	62.0	
Mid	SRPO†	53.1	21.9	0.0	25.0	+25.0
	UASRL	78.1	3.1	0.0	18.8	
High	SRPO†	92.7	2.7	0.0	4.5	+5.5
	UASRL	98.2	0.0	0.0	1.8	

Neither method changes a correct first answer into a wrong one in any group. UASRL's persistent errors (0→0) are similar to SRPO†'s in the low group and lower in the other two; 92.5% of them (98 of 106) fall in the low group, where the SFT model itself rarely answers correctly (Table 17).

Table 17: UASRL's persistent errors ($A_1 = A_2 =$ wrong) on the same 300 MathVista examples, by c_e group.

Group	Size	Persistent errors	Rate within group
Low	158	98	62.0%
Mid	32	6	18.8%
High	110	2	1.8%

C.5 Correction Given an Incorrect First Answer

Table 18 reports $P(A_2 \text{ correct} \mid A_1 \text{ incorrect})$ and, on MathVista, splits incorrect first answers into deferrals and attempts. Every correction on MathVista comes from a deferral: no method fixes a wrong attempted answer. The unconditional rate therefore measures how often a method defers, not how well it corrects. UASRL's lower rate reflects fewer deferrals; its deferred answers are resolved about as often as the other methods' (48.1% vs. 44.0% and 49.7%). On ScienceQA, UASRL corrects 20.2% of its wrong first answers, mostly in the inversion form (2.7 of the 4.0 points), while SRPO†'s high rate reflects its deferral of every answer.

C.6 Response Length and Early Stopping

Because A_1 is generated before reflection, stopping at the first `</answer>` tag yields a single-pass answer. Table 19 reports mean response length in words: on MathVista, stopping after UASRL's first answer shortens responses by 37% (98 vs. 156 words) while keeping $A_1 = 61.3$, compared with 49.9 for SRPO†.

Table 20 compares single-pass and full-pipeline accuracy. SRPO†'s single-pass A_1 is below Base on all three benchmarks. UASRL's exceeds Base on MathVista and ScienceQA and is closer to it on MMMU-Pro (-4.7 vs. -19.9 points).

Table 18: Conditional correction rate (%), $P(A_2 \text{ correct} \mid A_1 \text{ incorrect})$, with the number of incorrect first answers in parentheses. MathVista testmini (1,000) and ScienceQA test (2,000), Qwen2.5-VL-7B.

	MathVista			ScienceQA
Method	All	Deferred	Attempted	All
SFT	27.1 (451)	44.0 (277)	0.0 (174)	80.8 (1,253)
SRPO†	34.7 (501)	49.7 (350)	0.0 (151)	92.4 (1,795)
UASRL (ours)	16.8 (387)	48.1 (135)	0.0 (252)	20.2 (396)

Table 19: Mean response length (words) up to the first answer and for the full response. Ratio: full over first-answer mean length.

Dataset	Method	First answer	Full response	Ratio
MathVista	SFT	104	156	1.50×
MathVista	SRPO†	104	156	1.50×
MathVista	UASRL (ours)	98	156	1.59×
ScienceQA	SFT	33	78	2.36×
ScienceQA	SRPO†	32	78	2.44×
ScienceQA	UASRL (ours)	36	76	2.11×

C.7 Answer Changes

Table 21 reports how often the final answer differs from the first. On MathVista nearly every change replaces the placeholder; the SFT model also changes 10 correct first answers to wrong ones.

C.8 Cross-Domain and Out-of-Domain Evaluation

We evaluate Qwen2.5-VL-7B checkpoints beyond their training datasets, reporting both initial- and final-answer accuracy. These evaluations characterize transfer behavior separately from the main comparisons.

Cross-dataset transfer. Figure 6(a) evaluates UASRL in both transfer directions between MathVista and ScienceQA. Reflection improves accuracy for MathVista-to-ScienceQA transfer but leaves it nearly unchanged in the reverse direction. These results characterize direction-dependent reflection gains; without source-matched transfer baselines, they do not establish an advantage over SFT or SRPO.

Training source on a shared target. Figure 6(b) compares checkpoints on the same MMMU-Pro examples. Relative to MathVista-trained UASRL, the ScienceQA-trained checkpoint achieves higher A_1 but lower A_2. Stronger initial answers therefore need not imply stronger final answers under domain shift. The MathVista-trained SFT and SRPO checkpoints provide context for this training-source comparison.

MathVerse evaluation. Figure 6(c) extends evaluation to MathVerse (Zhang et al., 2024b). UASRL has slightly higher A_1 than SFT and SRPO, but does not exceed SRPO in A_2. Initial-answer accuracy remains low across all variants. This result limits extrapolation of the main findings to additional evaluation domains.

For MathVerse, decoding is greedy with a 1,024-token generation limit. The stored evaluator uses A_1 as A_2 when a second answer is absent and excludes responses without a parseable first answer. All plotted scores retain the saved accuracy summaries without rescoring. Each setting represents one evaluation run; no across-seed uncertainty is estimated.

Table 20: Single-pass (A_1) and full-pipeline (A_2) accuracy, Qwen2.5-VL-7B, in %; values from Table 2. ↑: higher is better; bold: best among trained methods.

	Single pass (A_1 ↑)			Full pipeline (A_2 ↑)		
Method	MathVista	ScienceQA	MMMU-Pro	MathVista	ScienceQA	MMMU-Pro
Base	57.4	77.4	41.1	–	–	–
SFT	54.9	37.4	27.6	66.1	81.5	47.7
SRPO†	49.9	10.3	21.2	67.3	82.9	47.0
UASRL (ours)	**61.3**	**80.2**	**36.4**	**67.8**	**84.2**	**49.1**

Table 21: Fraction (%) of responses whose final answer differs from the first, and, on ScienceQA, whose first option is the one following the final option. ↓: lower is better; bold: best.

	ScienceQA		MathVista
Method	Changed	First = next of final↓	Changed
SFT	61.4	60.6	28.7
SRPO†	100.0	97.0	34.9
UASRL (ours)	4.0	**2.7**	13.5

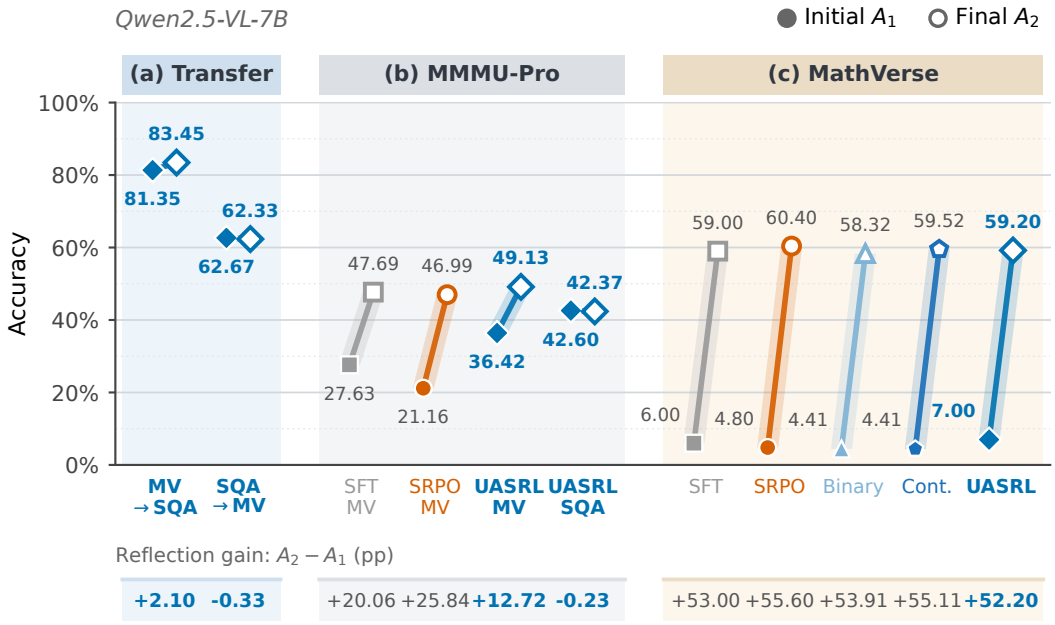


Figure 6: **Transfer beyond the training distribution.** Filled and open markers report A_1 and A_2 accuracy; connectors link the two scores for each checkpoint. (a) UASRL transfers between MathVista (MV) and ScienceQA (SQA), evaluating 2,000 ScienceQA and 300 MathVista examples. (b) All checkpoints share 1,730 MMMU-Pro examples in the standard four-option setting; the second label line identifies the training source. (c) MathVista-trained checkpoints on 500 MathVerse testmini examples. The binary and continuous capability-term variants each use 499 scored examples after one answer-parsing failure; other runs use 500. All regions share a 0–100% accuracy axis. The bottom row reports aggregate reflection gain, $A_2 - A_1$, in percentage points.