Warm Up with a Rubric, Then Merge: Grounding without Sacrificing Reasoning in Vision-Language Models

Anonymous ACL submission

Abstract

Process supervision improves perceptual grounding in RLVR-trained vision-language models, but it also incurs repeated verifier queries and reshapes the policy by suppressing exploratory entropy. We show that its benefit and cost are empirically separable: grounding gains emerge early, whereas the exploration cost—the policy's entropy held below that of outcome-only RLVR—accumulates with continued supervision. This hidden behavioral cost shows up as skill tradeoffs on static benchmarks and, more importantly, as a degraded initialization for subsequent outcome-only RLVR that prematurely limits its reachable ceiling. Motivated by this asymmetry, we propose *warmup-and-merge*: early-stop the process-supervised warmup, merge only a low-rank component of the warmup task vector into the base model, and continue with outcome-only RLVR under an adaptive on-policy reward that targets the model's current perceptual failures. Across multiple benchmarks and VLM backbones, our method improves perception while preserving exploration, achieving stronger final performance at a fraction of the verifier cost of continuous process supervision.

1 Introduction

Vision-language models (VLMs) trained with reinforcement learning from verifiable rewards (RLVR) have advanced rapidly on multimodal reasoning (Liu et al., 2025; Meng et al., 2025; Tan et al., 2025), yet these gains rest on a fragile foundation: a model that never grounds its rationale in the image can still answer correctly by leaning on language priors or dataset regularities (Li et al., 2025; Luo et al., 2025; Wan et al., 2024). To close this gap, recent work adds *process supervision*—rewards that score perception and grounding rather than only the final answer—and reports consistent perceptual gains (Xiao et al., 2025; Li et al., 2025; Wang et al., 2026). Such rewards query a verifier on every sampled rollout, so their cost grows with the number of supervised steps. Less understood is a second effect: how process supervision reshapes the policy itself, beyond the accuracy it reports.

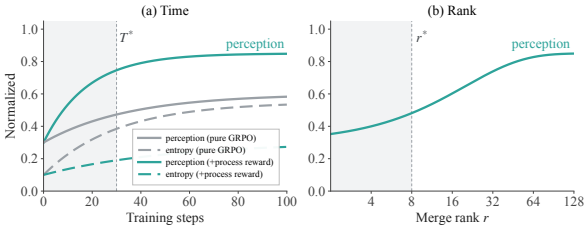


Figure 1: **Perception vs. entropy under two axes. (a) Time.** Pass-rate (solid) and policy entropy (dashed) along training, with the process reward (green) and a pure GRPO baseline (gray). The shaded band marks the warmup phase $[0, T^*]$; the two entropy curves share the same SFT initialization. **(b) Rank.** The same two quantities as a function of the model merge rank r; indicating a low-rank perception update. All values are normalized to $[0, 1]$.

We study process supervision from a different angle: rather than asking directly how to build a stronger verifier, we ask *when* and *where* its effect enters the model—and find that its benefit and its cost are *separable*. The benefit is perceptual: visual grounding improves and saturates early. The cost is behavioral: process supervision suppresses the policy's exploration, holding its entropy *below* that of outcome-only RLVR throughout training and pulling it toward a narrow, verifier-preferred mode—a manifestation, in the multimodal process-reward setting, of the limited entropy budget that governs RL more broadly (Cui et al., 2025). These two effects come apart along two axes. In *time*, grounding saturates within an early window, while the entropy gap relative to outcome-only training persists with further supervision. In *parameter space*, they separate by rank: merging only a low-rank component of the warmup update into the base model retains the grounding gain, whereas a

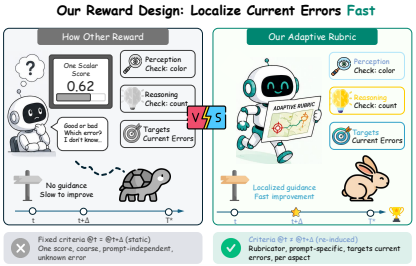


Figure 2: Comparison of reward strategies. The baseline uses a fixed scalar score with coarse feedback, while our adaptive rubric localizes current errors and dynamically adjusts criteria for faster, targeted learning.

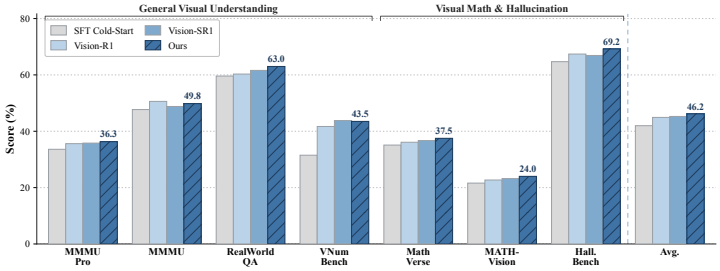


Figure 3: Performance comparison across multimodal benchmarks. Ours consistently outperforms baselines on both perception-oriented (e.g., RealWorldQA, MMMU) and reasoning-intensive tasks (e.g., MathVerse, MATH-Vision), achieving the highest average score.

higher-rank merge imports the exploration cost and underperforms even a cold start. Reconstructing the update at low rank thus transfers grounding while leaving most of that cost behind.

This suppressed exploration is not benign. Already on static evaluation it trades capability across skills: process supervision improves grounding (e.g., HallusionBench) while *lowering* accuracy on reasoning-oriented benchmarks (e.g., MMStar) relative to outcome-only RLVR. Its larger cost is paid later: reasoning gains under RL are bought with exploration, so a policy whose exploration is spent too early saturates before its reasoning is fully developed (Cui et al., 2025). Running process supervision to convergence is therefore self-limiting rather than free—its attainable ceiling falling below what exploration-preserving training reaches.

This asymmetry motivates a direct strategy: extract the perceptual contribution once and discard the entropic cost. We instantiate it as *warmup-and-merge*, separating the two effects along the very axes on which they diverge. In time, process supervision is applied only as a short, early-stopped warmup, halted before exploration has been substantially suppressed. In parameter space, we decompose the resulting warmup task vector and merge only its low-rank component—which carries the perceptual gain—into the base model, leaving behind the higher-rank directions that carry the exploration cost. Standard outcome-only RLVR then proceeds from this initialization, which is perceptually stronger than the base model yet retains the entropy that exploration requires. Process supervision thereby supplies a perceptual prior once, up front, rather than an ongoing signal whose side effect must be continually counterbalanced.

The short warmup, in turn, constrains the reward. With only a few update steps and no room to correct errors gradually, the reward must localize the policy's current perceptual errors quickly; what matters is not the verifier's strength but how directly it pinpoints those errors. Existing process rewards fit this constraint poorly. Fixed-annotation rewards (Xiao et al., 2025) compare rollouts against a predefined attribute set and yield little signal once it is satisfied, while self-evaluation rewards (Li et al., 2025) emit a single scalar that marks a rollout as wrong without indicating which step to revise. We instead induce the reward criteria *on-policy*: at each step we derive them from the contrast between the current rollouts and a fixed teacher reference, re-inducing them as the policy shifts so that supervision tracks the model's present failures. We further keep perceptual and reasoning criteria as separate reward terms rather than pooling them into one scalar, preventing the perceptual signal from being diluted. Consistent with the asymmetry above, the design helps most when the warmup is shortest: its advantage over fixed rewards is largest at small step budgets and narrows as the window grows.

Our contributions are as follows:

- We identify a *two-fold asymmetry* in process-supervised RLVR for VLMs: the perceptual benefit appears early and is retained by merging only a *low-rank* component of the warmup update, whereas its cost—suppressed exploration, with policy entropy held *below* outcome-only RLVR—accumulates with continued supervision and surfaces both as degraded reasoning-type accuracy and as a weaker initialization for subsequent outcome-only RL.

- Guided by this asymmetry, we propose *warmup-and-merge*: a short, early-stopped process-supervised warmup driven by an adaptive on-policy *rubric reward* that localizes the model's current perceptual errors, a low-rank merge of the warmup task vector into the base model, and outcome-only RLVR thereafter with no further verifier queries.

- Across multiple benchmarks and two VLM backbones, our method improves perception while preserving exploratory entropy, achieving higher final performance at a fraction of the verifier cost of continuous process supervision.

2 Related Works

Process supervision in VLM-RLVR. RLVR, especially GRPO/R1-style optimization (Shao et al., 2024; DeepSeek-AI, 2025), has become a common recipe for improving multimodal reasoning in VLMs (Liu et al., 2025; Meng et al., 2025; Tan et al., 2025; Huang et al., 2025; Shen et al., 2025). However, outcome-only rewards may be satisfied by answers that exploit language priors or dataset shortcuts without grounding in the image. Recent work therefore adds process or intermediate supervision for perception and grounding: Perception-R1 (Xiao et al., 2025) scores rationales against visual annotations; Vision-SR1 (Li et al., 2025) checks whether generated perception suffices for answering; and R1-VL (Zhang et al., 2025) uses dense step-wise reasoning rewards. These methods improve perceptual accuracy, but mainly focus on designing the supervision signal. In contrast, we study how process supervision reshapes the model over training time and parameter space, showing that its perceptual benefit and its effect on reasoning are separable.

Training dynamics and capability interference in RL fine-tuning. A growing line of work examines what RL fine-tuning changes in a model beyond final accuracy, including capability forgetting and alignment tax (Lin et al., 2024; Li et al., 2024), the narrowing or sharpening of output distributions (Kirk et al., 2024; O'Mahony et al., 2024; Xiao et al., 2024), and trade-offs between competing skills during multi-objective training (Guo et al., 2024; Wang et al., 2024a). In the multimodal setting, improving perception and preserving reasoning can pull in different directions. Our work contributes a specific characterization of this interference under process supervision: perception gains saturate early while reasoning degrades with continued training, and the two effects occupy distinct rank subspaces of the parameter update. This connects capability trade-offs to the geometry of the update, relating process-supervised RL to parameter-space views of adaptation such as low-rank updates and task vectors (Hu et al., 2022; Ilharco et al., 2023), rather than to accuracy alone.

3 Method

3.1 Preliminaries

RLVR with GRPO. We optimize a VLM policy π_θ with GRPO (Shao et al., 2024). For a multimodal prompt $x = (v, q)$ pairing an image v with a query q, the policy samples a group of G rollouts $\{y_i\}_{i=1}^{G}$, each assigned a reward r_i. GRPO computes the group-normalized advantage $\hat{A}_i = (r_i - \mu)/\sigma$, where μ, σ are the mean and standard deviation of $\{r_i\}$, and updates π_θ with a clipped surrogate objective and a KL penalty to a frozen reference π_{ref}. An *outcome* reward r^{out} scores only final-answer correctness; a *process* reward r^{proc} additionally evaluates perception and grounding and requires querying a verifier on every rollout. A process-supervised step uses $r = r^{\text{out}} + \lambda\, r^{\text{proc}}$.

Task vector. Following (Ilharco et al., 2023), for a model fine-tuned from base parameters θ_0 to θ, the *task vector* is the parameter delta $\Delta\theta = \theta - \theta_0$. For a weight matrix with delta $\Delta W \in \mathbb{R}^{m\times n}$ and SVD $\Delta W = U\Sigma V^\top$, the *rank-$r$ reconstruction* is $\Delta W_{\le r} = U_{:,1:r}\, \Sigma_{1:r,1:r}\, V_{:,1:r}^\top$. We write $\Delta\theta_{\le r}$ for the collection of rank-r reconstructions over all weight matrices.

3.2 Analysis: From Asymmetry to Design

Setup. We localize *when* process supervision helps and hurts by running a process reward (Vision-SR1 (Li et al., 2025)) against an outcome-only GRPO baseline on Qwen2.5-VL-7B, tracking held-out validation accuracy on a grounding (HallusionBench) and a reasoning-oriented (MM-Star) benchmark, together with token-level policy entropy (full-vocabulary, on each model's own rollouts).

Process supervision trades exploration for grounding. Figure 5 shows the trade-off. Grounding (HallusionBench) improves and

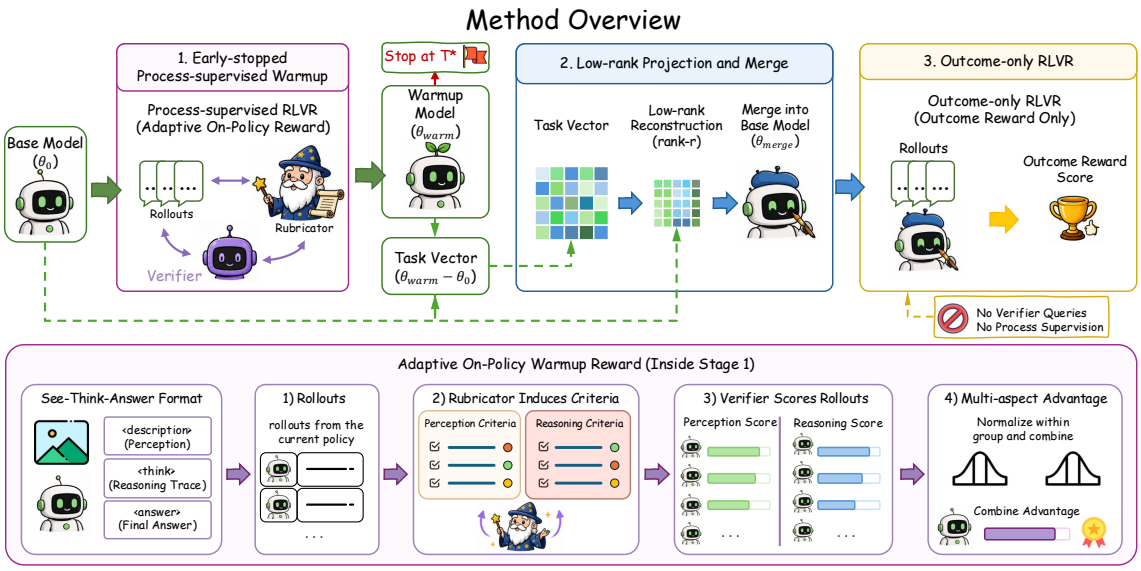


Figure 4: **Method overview.** Our pipeline has three stages: **(1)** a short, early-stopped process-supervised RLVR warmup with an *adaptive on-policy reward* yields θ_{warm}; **(2)** the task vector $\theta_{\text{warm}} - \theta_0$ is SVD-projected to rank r and merged into the base model as θ_{merge}, retaining the low-rank perceptual gain while removing high-rank directions that degrade reasoning; **(3)** *outcome-only RLVR* continues from θ_{merge} without any further verifier queries. *Bottom:* the Stage-1 adaptive reward—a Rubricator induces prompt-specific perception/reasoning criteria from See–Think–Answer rollouts, a Verifier scores them, and the normalized scores form a multi-aspect advantage.

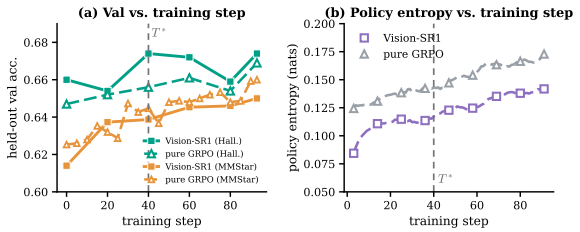


Figure 5: **Process supervision trades exploration for grounding.** **(a)** Held-out validation along training for a process reward (Vision-SR1) vs. outcome-only GRPO: process is higher on grounding (HallusionBench, teal) but lower on the reasoning-oriented benchmark (MMStar, orange). **(b)** Its policy entropy stays below outcome-only RLVR throughout, indicating suppressed exploration. T^* marks the early-stop point.

saturates within the early steps, while accuracy on the reasoning-oriented benchmark (MMStar) sits below outcome-only RLVR; in parallel, policy entropy stays below outcome-only RLVR throughout. Process supervision thus buys an early perceptual gain by suppressing exploration—a cost that accumulates with continued training, motivating an *early stop* that keeps the grounding before the exploration loss compounds.

From asymmetry to design. This argues for *separating* the two effects rather than trading them off: keep the early perceptual gain and discard the rest. We do so by merging only a *low-rank* component of the warmup update into the base model; §5.1 shows that this low-rank merge retains the benefit downstream, whereas a high-rank merge imports the cost and ends up below even a cold start.

3.3 Warmup-and-Merge

Our method proceeds in three stages.

(1) Early-stopped process-supervised warmup. Starting from θ_0, we run full-parameter process-supervised RLVR with the adaptive reward of §3.4, stopping at T^*. This yields θ_{warm} and the task vector $\Delta\theta = \theta_{\text{warm}} - \theta_0$.

(2) Low-rank projection and merge. We apply the rank-r reconstruction to the task vector of *every* weight matrix and merge the result into the base model:

$$\theta_{\text{merge}} = \theta_0 + \Delta\theta_{\leq r}.$$

This retains the low-rank perceptual gain while discarding the higher-rank directions that carry reasoning damage. The rank r is selected on a held-out validation set as the smallest value that retains perception without degrading reasoning; the same r is applied across all matrices.

(3) Outcome-only RLVR. From the merged initialization θ_{merge}, we continue with standard RLVR using only the outcome reward r^{out}, with no further verifier queries. Process supervision thus contributes a one-time perceptual prior, and the long phase of training incurs no recurring verifier cost.

3.4 Adaptive On-Policy Warmup Reward

The short warmup budget imposes a specific requirement on the reward signal: with only a few hundred steps before T^*, the reward must localize the model's current errors rather than rely on a long training horizon to correct them gradually. Existing process rewards do not satisfy this requirement. A fixed-annotation reward (Xiao et al., 2025) scores rollouts against a predefined set of visual attributes and ceases to provide informative signal once these are satisfied, while a single self-evaluation mechanism (Li et al., 2025) yields a coarse, prompt-independent signal that does not indicate which error to correct.

We instead induce the reward criteria on-policy. At each step, the Rubricator contrasts the current policy's rollouts with a teacher reference and induces criteria that characterize the errors present in those rollouts. As the policy evolves, the induced criteria adapt accordingly, keeping supervision aligned with the model's current failures throughout the short window—a property that fixed or non-inductive rewards do not provide.

Teacher references. The induction requires a reference of the desired output for each prompt. We construct two references offline with a stronger teacher VLM: a *perception reference*, obtained by prompting the teacher to describe the image content relevant to the question, and a *reasoning reference*, obtained by prompting it to produce a step-by-step solution. These references are fixed throughout training and serve only as a contrast against which the current policy's rollouts are compared; the criteria themselves are induced on-policy from this contrast, so that what is supervised still tracks the model's current errors rather than the teacher's outputs directly.

Criterion induction and scoring. We follow the Vision-SR1 (Li et al., 2025) and adopt the See-Think-Answer rollout format, which separates a perceptual statement (`<description>`) from a reasoning trace (`<think>` and the final answer), and induce criteria separately for the two aspects. Given a prompt (v, q) and the group of G rollouts $\{y_i\}_{i=1}^{G}$, the Rubricator contrasts the rollouts against two teacher references—a perception reference and a reasoning reference—and induces, for each aspect $a \in \{\text{perc}, \text{reason}\}$, a set of K_a weighted criteria $\{(c_k^a, w_k^a)\}_{k=1}^{K_a}$, where each $w_k^a > 0$ denotes the importance of criterion c_k^a. A Verifier then scores each rollout against these criteria, giving a per-aspect pass rate

$$s^a(y_i) = \frac{\sum_{k=1}^{K_a} w_k^a \, \mathbb{1}[\, y_i \text{ satisfies } c_k^a \,]}{\sum_{k=1}^{K_a} w_k^a}.$$

Inducing criteria from the group of rollouts at each step yields supervision specific to the current prompt and the model's present behavior, in contrast to a reward defined over a fixed criterion set.

Multi-aspect advantage. Rather than collapsing the two aspects into a single scalar reward, we retain them as separate components and normalize each within its group before combining, following a multi-reward policy optimization scheme (Li et al., 2025). For aspect a, the group-normalized advantage is $\hat{A}_i^a = (s^a(y_i) - \mu^a)/\sigma^a$, where μ^a and σ^a are the mean and standard deviation of $\{s^a(y_j)\}_{j=1}^{n}$, and the total advantage is the weighted sum $\hat{A}_i = \sum_a \lambda_a \hat{A}_i^a$ with $\lambda_{\text{perc}} = \lambda_{\text{reason}} = 0.5$. Keeping the two aspects separate preserves the perceptual signal that the warmup is intended to strengthen, rather than allowing it to be diluted in a pooled reward.

4 Experiments

Setup. We use Qwen2.5-VL-3B and -7B (Yang et al., 2025) as base VLMs. The warmup are trained on Vision-SR1-5.9K and the RLVR stage train on Vision-SR1-47K (Li et al., 2025) ($\sim$47K examples from 24 open-source benchmarks spanning math, science, and general visual understanding). Following Vision-SR1 (Li et al., 2025), we evaluate general understanding on MMMU (Yue et al., 2024), MMMU-Pro (Yue et al., 2025), RealWorldQA (xAI, 2024), and VisNumBench (Weng et al., 2025), math reasoning on MathVerse (Zhang et al., 2024) and MATH-Vision (Wang et al., 2024b), and hallucination on HallusionBench (Guan et al., 2024), reporting accuracy with Qwen2.5-14B-Instruct (Yang et al., 2025) as the judge for non-multiple-choice items. We compare against the zero-shot base model, outcome-only RLVR (Vision-R1 (Huang et al.,

Table 1: Performance comparison of OURS with representative vision-language reasoning baselines across general visual understanding, visual math, and hallucination benchmarks. RW: RealWorldQA; VNum: VisNumBench; Hall.: HallusionBench. Higher scores indicate better performance

Methods	**General Visual Understanding**				**Visual Math & Hallucination**			**Avg.**
	MMMU Pro	**MMMU**	**RW QA**	**VNum Bench**	**Math Verse**	**MATH-Vision**	**Hall. Bench**	
Visionary-R1 (3B) (Xia et al., 2025)	27.4	30.6	56.9	10.0	-	40.4	26.7	33.9
Perception-R1 (7B) (Xiao et al., 2025)	36.8	40.9	69.4	15.9	54.3	28.6	65.4	45.2
Vision-R1 (7B) (Huang et al., 2025)	34.9	42.8	60.1	33.0	52.4	-	32.2	44.5
Backbone model: Qwen2.5-VL-3B								
SFT Cold-Start	33.6	47.7	59.6	31.5	35.1	21.6	64.7	42.0
Vision-R1 (Huang et al., 2025)	35.6	47.6	60.3	41.7	36.1	22.7	66.9	44.4
Vision-SR1 (Li et al., 2025)	35.8	48.8	61.6	**43.8**	36.7	23.2	67.4	45.3
OURS-**Warmup**	34.8	46.4	61.2	39.8	34.2	16.7	66.8	42.8
OURS-**RLVR**	**36.3**	**49.8**	**63.0**	43.5	**37.5**	**24.0**	**69.2**	**46.2**
Backbone model: Qwen2.5-VL-7B								
SFT Cold-Start	33.4	47.7	63.8	43.5	42.0	26.3	66.7	46.2
Vision-R1 (Huang et al., 2025)	36.2	51.6	63.5	43.0	44.0	28.7	66.6	47.7
Vision-SR1 (Li et al., 2025)	42.5	**57.8**	69.0	43.5	44.4	27.2	71.1	50.8
OURS-**Warmup**	37.2	50.8	68.2	44.7	46.6	29.5	69.3	49.5
OURS-**RLVR**	**43.4**	53.5	**70.3**	**45.4**	**47.0**	**30.5**	**72.0**	**51.7**

2025)), continuous, and self-rewarding Vision-SR1 (Li et al., 2025). For a controlled comparison, all baselines are re-trained and re-evaluated within the Vision-SR1 codebase, under identical data, prompts, and evaluation protocol as our method.

Training. We optimize with GRPO (Shao et al., 2024); the full RLVR run is 93 steps (one epoch over Vision-SR1-47K at rollout batch size 512). The warmup is full-parameter and early-stopped at the validation understanding-saturation point $T^* = 20$ (§3.2); a rank-r component of the warmup update is then merged into the base model, with r chosen on validation. We adopt the multi-reward scheme of Vision-SR1 (Li et al., 2025) ($\lambda_{\text{perc}}=\lambda_{\text{reason}}=0.5$). Training uses 8× NVIDIA H20 (96 GB) GPUs; full hyperparameters are in App. A.

4.1 Main Results

Table 1 reports accuracy across understanding, visual-math, and hallucination benchmarks. OURS-RLVR attains the best average at both scales—46.2 (3B) and 51.7 (7B)—ahead of the strongest baseline, the continuously process-supervised Vision-SR1 (45.3/50.8), and of outcome-only RLVR (44.9/47.7).

Grounding improves without sacrificing reasoning. Our gains are largest on hallucination (HallusionBench: 69.2/72.0, best at both scales) and on visual math, where OURS-RLVR is also the strongest—so the improved visual grounding does not trade off against reasoning.

A short warmup beats supervision throughout. Vision-SR1 applies process supervision at every step; our method uses it only during a brief warmup, yet matches Vision-SR1 on general understanding while improving visual math (by up to +∼3 points at 7B) and hallucination—consistent with the exploration cost of continuous supervision identified in §3.2, at a fraction of the verifier queries.

The warmup is an intermediate, not the product. OURS-Warmup alone is weaker than the RLVR baselines (e.g., 42.8 at 3B); the low-rank merge followed by outcome-only RLVR recovers and surpasses them, confirming that the perceptual prior pays off only after the subsequent RL stage.

4.2 Ablations

We ablate three pipeline choices in order—the warmup checkpoint T^*, the merge rank r, and the warmup reward—each run through the same outcome-only RLVR stage, with T^* and r chosen on held-out validation (Table 2).

Warmup checkpoint T^*. Performance is unimodal in T^*: an earlier checkpoint underfits the perceptual gain (45.5) and a later one (45.7) carries more of the exploration cost, with continuous supervision worst (44.5, below even outcome-only

Table 2: Full per-benchmark ablation results, reporting individual benchmark scores under each ablation. RW: RealWorldQA; VNum: VisNumBench; Hall.: HallusionBench. Each variant is trained through the complete pipeline; we report final accuracy after outcome-only RLVR

Variant	General Visual Understanding				Visual Math & Hallucination			Avg.
	MMMU Pro	MMMU	RW QA	VNum Bench	Math Verse	MATH-Vision	Hall. Bench	
SFT Cold-Start (Qwen2.5-VL-3B)	33.6	47.7	59.6	31.5	35.1	21.6	64.7	42.0
Outcome-only RLVR	35.6	47.6	60.3	41.7	36.1	22.7	66.9	44.4
OURS-**RLVR** (T=20*, r=32*)	**36.3**	**49.8**	**63.0**	**43.5**	**37.5**	**24.0**	**69.2**	**46.2**
(1) Warmup checkpoint T								
earlier ckpt (T=5)	35.7	49.1	61.9	42.2	37.2	23.7	68.5	45.5
later ckpt (T=45)	36.0	49.5	62.6	43.1	36.7	23.0	68.7	45.7
(2) Projection rank r								
r=8	35.6	49.1	62.0	42.2	37.2	23.8	68.7	45.5
r=full	36.1	48.6	62.8	43.1	34.6	20.8	68.3	44.9
(3) Warmup reward								
fixed reward	35.5	49.2	61.8	42.0	37.1	23.6	68.6	45.4

RLVR). T^* takes the understanding gain before the cost compounds.

Merge rank r. Accuracy peaks at an intermediate r^* (46.2). A much smaller rank (r=8) underfits the perceptual prior (45.5), whereas a full-rank merge erases the gain altogether, dropping to 44.9—matching outcome-only RLVR with no merge. Only a low-rank merge turns the warmup into a net improvement; importing the higher-rank directions reinstates the cost that cancels it, so we select the smallest validation-optimal rank.

Warmup reward. At a matched short budget, the adaptive reward beats a fixed-criterion one (46.2 vs. 45.4), yielding a stronger low-rank component to merge—specific to the short-warmup regime.

5 Analysis

5.1 Does the Merged Initialization Survive Outcome-Only RLVR?

To test whether the perceptual prior survives once process supervision is removed, we run identical outcome-only RLVR from a low-rank (r=32) and a high-rank (r=256) merge of the warmup update, against a cold start θ_0 and a self-rewarding run (Li et al., 2025) (Fig. 6). Both merges start around 64.2–64.4, above the cold start (62.5) and self-rewarding (61.4), so merging transfers a usable prior before any update. The low-rank merge then builds on this head start to finish highest at 67.0 (66.0 outcome-only, 65.0 self-rewarding), whereas the high-rank merge plateaus and ends at 63.8, *below* even the cold start. Merging thus helps only at low rank; the higher-rank components squander the head start.

Figure 6: **The low-rank merge gives a perceptual head start that survives outcome-only RLVR.** Held-out MMStar accuracy over training for self-rewarding supervision (Li et al., 2025), outcome-only RLVR from a cold start, and ours (rubric warmup + low-rank merge, then outcome-only RLVR). The merged run leads at initialization (64.2 vs. 62.5), is briefly matched near step 40, and is highest at the final checkpoint. Faint lines are raw values, bold lines EMA-smoothed; y-axis truncated to 0.61–0.67.

5.2 Cost Analysis

A further, practical consequence of stopping early is cost. Process supervision queries a verifier on every rollout, so its cumulative cost grows linearly with training length. Our method reduces total

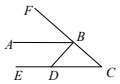
verifier queries by $N\times$ relative to continuous supervision while avoiding its reasoning degradation, turning a recurring per-step cost into a one-time one. This is the practical counterpart of the time-dimension asymmetry: there is no benefit, and a recurring cost, to extending process supervision past T^*.

5.3 What the Warmup Changes: More Faithful Reasoning

Table 3: **Language Shortcut Rate (LSR, %; lower is better)** on three benchmarks. A lower LSR means fewer correct answers reached through a language shortcut despite incorrect visual perception. RW: RealWorldQA. Judged by Gemini-2.5-Flash via the two-step self-containment check (§5.3).

Method	MMMU	RW	MathVerse	Avg.
Qwen2.5-VL-3B				
Vision-SR1	6.3	10.8	10.3	9.1
Ours	**4.7**	**7.9**	**6.8**	**6.5**
Qwen2.5-VL-7B				
Vision-SR1	6.5	13.4	11.5	10.5
Ours	**5.0**	**9.2**	**7.9**	**7.4**

Case study (MMStar val): same question, two reward paradigms



Q: As shown, AB ∥ CD, BD ⊥ CF (foot at B), ∠ABF = 35°; find ∠BDC.
(A) 25° (B) 35° (C) 45° (D) 55° [GT: D (55°)]

Self-rewarding : answer B (wrong, score 0.10)
Description: The image shows two parallel lines AB and CD. There is a line segment BD that intersects CD at point D. There is also a line segment BF that intersects AB at point B. BD is perpendicular to CF at point B. Angle ABF is given as 35 degrees. We need to find the measure of angle BDC.
Reasoning (think): Since AB is parallel to CD, we know that angle ABF and angle DBC are alternate interior angles, so they are equal. Therefore, angle DBC = angle ABF = 35 degrees. Since BD is perpendicular to CF, angle DBC is a right angle, but we are only given that BD is perpendicular to CF. This means that angle DBC is not a right angle. However, we are given that angle ABF is 35 degrees. Since AB is parallel to CD, angle BDC is equal to angle ABD. Also, since BD is perpendicular to CF, angle FBD is 90 degrees. Therefore, angle ABD = angle FBD − angle ABF = 90 − 35 = 55 degrees. Since AB is parallel to CD, angle BDC is equal to angle ABD, which is 55 degrees.
Alternatively, since AB ∥ CD, we have ABF = BDC because they are alternate interior angles. Also, since BD ⊥ CF, ∠FBD = 90°. We are given ∠ABF = 35°. Therefore, ∠BDC = ∠ABF = 35°.

Rubric process reward (ours) : answer D (correct, score 1.00)
Description: The image shows two parallel lines AB and CD. A line segment BD connects the two parallel lines. Another line segment CF intersects BD at point B, and BD is perpendicular to CF, meaning angle FBD is 90 degrees. The angle ABF is labeled as 35 degrees. The question asks for the measure of angle BDC.
Reasoning (think): Since AB is parallel to CD, we know that angle ABF and angle D are corresponding angles, so they are equal. Therefore, ∠D = ∠ABF = 35 degrees. Also, since BD is perpendicular to CF, ∠DBC is 90 degrees. We want to find ∠BDC. In triangle BDC, we know ∠DBC = 90 degrees and ∠D = 35 degrees. The sum of the angles in a triangle is 180 degrees.
Therefore, ∠BDC = 180 − 90 − 35 = 55 degrees. So, the answer is 55 degrees.

Figure 7: Case study (MMStar val): Comparison of self-rewarding and rubric process–reward models on a geometry problem. The self-rewarding model predicts 35°, while the rubric process–reward model correctly finds 55° using geometric reasoning.

Beyond aggregate scores, we ask how our reward reshapes the model's reasoning. Quantitatively, we measure the Language Shortcut Rate (LSR; Table 3), the fraction of correct answers reached *despite* an incorrect visual reading—a high LSR means the answer was obtained through a language shortcut rather than genuine visual reasoning. Our model attains a lower LSR than the self-rewarding baseline (Li et al., 2025) on every benchmark (vs. on average), indicating that its correct answers rest on faithful reasoning over the image rather than on priors. Qualitatively (Fig. 8; more in App. C), the two rewards perceive the scene comparably but reason differently: on the geometry example, the self-rewarding model conflates the angle relations and settles on the wrong value, whereas our model carries out the correct triangle–angle-sum derivation and reaches the right answer. The warmup thus changes *how the model reasons over the image*, not merely what it perceives.

6 Conclusion

We revisit process-supervised reinforcement learning for vision–language models from a mechanistic perspective, showing that its perceptual benefits and behavioral costs are separable across time and parameter subspaces. Process supervision improves grounding early, but continued training collapses policy entropy and harms downstream reasoning. Our proposed *warmup-and-merge* exploits this asymmetry: a short process-supervised warmup captures perceptual gains, low-rank merging preserves them without importing entropy collapse, and subsequent outcome-only RLVR maintains exploration capacity. Empirically, this strategy restores reasoning performance while retaining visual understanding, yielding higher final accuracy with substantially lower verifier cost. More broadly, our results suggest that process-level rewards shape distinct functional subspaces, only some of which should be retained.

Limitations

This work has several limitations. First, our method relies on validation-based choices of the warmup checkpoint T^* and projection rank r, which may require retuning across model families, datasets, and reward formulations. Second, low-rank merging intentionally filters update directions to preserve early visual-understanding gains, but it may also discard useful high-rank information, especially under longer warmup, stronger supervision, or more diverse data. Finally, our explanation of why low-rank components preserve perceptual gains remains empirical, and the method does not fully address broader challenges of outcome-based optimization such as shortcut learning, or reward hacking.

References

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1454. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *International Conference on Learning Representations*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations*.

Shengzhi Li, Rongyu Lin, and Shichao Pei. 2024. Multimodal preference alignment remedies degradation of visual instruction tuning on language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 14188–14200.

Zongxia Li, Wenhao Yu, Chengsong Huang, Rui Liu, Zhenwen Liang, Fuxiao Liu, Jingxi Che, Dian Yu, Jordan Boyd-Graber, Haitao Mi, and Dong Yu. 2025. Self-rewarding vision-language model via reasoning decomposition. *arXiv preprint arXiv:2508.19652*.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of rlhf. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*.

Tiange Luo, Ang Cao, Gunhee Lee, Justin Johnson, and Honglak Lee. 2025. Probing visual language priors in VLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41120–41156. PMLR.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. 2024. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Xiansheng Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. 2025. Reason-rft: Reinforcement fine-tuning for visual reasoning of vision language models. *arXiv preprint arXiv:2503.20752*.

David Wan, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. 2024. Contrastive region guidance: Improving grounding in vision-language models without training. In *European Conference on Computer Vision*.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, Alexandre Rame, Johan Ferret, Geoffrey Cideron, Le Hou, Hongkun Yu, Amr Ahmed, Aranyak Mehta, Leonard Hussenot, Olivier Bachem, and Edouard Leurent. 2024a. Conditional language policy: A general framework for steerable multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2153–2186.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024b. Measuring multimodal mathematical reasoning with MATH-vision dataset. In *Advances in Neural Information Processing Systems*, volume 37.

Sudong Wang, Weiquan Huang, Xiaomin Yu, Zuhao Yang, Hehai Lin, Keming Wu, Chaojun Xiao, Chen Chen, Wenxuan Wang, Beier Zhu, Yunjian Zhang, and Chengwei Qin. 2026. Beyond sft-to-rl: Pre-alignment via black-box on-policy distillation for multimodal rl. *arXiv preprint arXiv:2604.28123*.

Tengjin Weng, Jingyi Wang, Wenhao Jiang, and Zhong Ming. 2025. VisNumBench: Evaluating number sense of multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

xAI. 2024. Grok-1.5 vision preview. https://x.ai/news/grok-1.5v. Introduces the RealWorldQA benchmark.

Jiaer Xia, Yuhang Zang, Peng Gao, Yixuan Li, and Kaiyang Zhou. 2025. Visionary-r1: Mitigating shortcuts in visual reasoning with reinforcement learning. *arXiv preprint arXiv:2505.14677*.

Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J. Su. 2024. On the algorithmic bias of aligning large language models with rlhf: Preference collapse and matching regularization. *arXiv preprint arXiv:2405.16455*.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. 2025. Advancing multimodal reasoning capabilities of multimodal large language models via visual perception reward. *arXiv preprint arXiv:2506.07218*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jingren Yang, Jianxin Zhou, and Junyang Lin. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2025. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, Vienna, Austria. Association for Computational Linguistics.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. 2024. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

A Implementation Details

A.1 Training Configuration

Table 4: Stage 1 (Rubric process-reward warmup) training hyperparameters.

Hyperparameter	Value
Base model	`Qwen-2.5VL-7B-Cold-Start`
Train data	`Vision-SR1 5.9k`
Reward	`self-reward + rubric process`
Rubric judge	`gpt-4.1`
Learning rate	`1e-6`
LR scheduler	`constant`
Warmup steps	`0`
Training epochs	`5`
Temperature	`1.0`
Clip range ϵ	`0.2`
Clip range high ϵ_{high}	`0.3`
Precision	`bf16`
Max grad norm	`1.0`
Iterations per run (PPO epochs)	`1`
KL coefficient β	`0.01`
Max prompt length	`12800`
Max completion length	`4096`
Per-device train batch	`1`
Gradient accumulation steps	`16`
Num generations	`8`
Rollout batch size	`512`

A.2 Process Reward: Rubricator and Verifier

Models. Both the Rubricator and the Verifier are instantiated by a single frozen LLM, `GPT-4.1`, queried through an OpenAI-compatible endpoint at temperature 0. For each training prompt we issue exactly *two* calls: one to the Rubricator to induce a rubric, and one to the Verifier to grade all on-policy rollouts of that prompt against it. The two calls are text-only; the Rubricator is *not* shown the raw image (see below), and the Verifier never sees the gold answer, so neither model can shortcut the judgement.

Reference responses. The reference signal comes from the teacher trajectories of our converted PRISM-5.9k corpus. Each example carries a teacher *caption* $c^\star$ (a perception reference: what is in the image) and a teacher *chain-of-thought* frozen LLM, `GPT-4.1`, queried through an OpenAI-compatible endpoint at temperature 0. For each training prompt we issue exactly *two* calls: one to the Rubricator to induce a rubric, and one to the Verifier to grade all on-policy rollouts of that prompt against it. The two calls are text-only; the Rubricator is *not* shown the raw image (see below), and the Verifier never sees the gold answer, so neither model can shortcut the judgement.

Table 5: Stage 2 (RLVR) training hyperparameters.

Hyperparameter	Value
Base model	`stage 1 model`
Train data	`Vision-SR1-47K`
Reward	`accuracy (outcome-only)`
Learning rate	`1e-6`
LR scheduler	`constant`
Warmup steps	`0`
Training epochs	`1`
Temperature	`1.0`
Clip range ϵ	`0.2`
Clip range high ϵ_{high}	`0.3`
Precision	`bf16`
Max grad norm	`1.0`
Iterations per run (PPO epochs)	`1`
KL coefficient β	`0.01`
Max prompt length	`12800`
Max completion length	`4096`
Per-device train batch	`1`
Gradient accumulation steps	`16`
Num generations	`8`
Rollout batch size	`512`

Rubric induction. Given $(q, c^\star, \text{cot}^\star, a^\star)$ and the set of student responses, the Rubricator produces a rubric with two fixed categories, $\mathcal{C} = \{\text{PERCEPTION}, \text{REASONING}\}$. For each category $c \in \mathcal{C}$ it induces $m_c \in [3, 6]$ concrete, binary-checkable criteria, each carrying an integer weight ("points") $p_{c,k} \in \{1, \ldots, 5\}$, $k = 1, \ldots, m_c$. The points are deliberately *sharpened* rather than spread evenly: discriminative criteria receive 4–5 points while surface/format requirements are capped at 1. The perception criteria check visual grounding (objects, counts, spatial relations, labels/text, values read off the image) and self-containedness, and crucially *forbid* any criterion that rewards stating the final answer inside the description, preventing answer leakage into the perception channel. The reasoning criteria check the logical steps and final-answer correctness. Criteria are induced per prompt, so the rubric adapts to each question (e.g. counting vs. geometry vs. chart reading).

Verification and weighted pass-rate. The Verifier scores every rollout i of the prompt in a single call, emitting a binary judgement $j^i_{c,k} \in \{0, 1\}$ for each criterion. We *recompute* each category score from the judgements and points (rather than trusting the model's own arithmetic),

$$s^i_c = \sum_{k=1}^{m_c} p_{c,k}\, j^i_{c,k}, \qquad S_c = \sum_{k=1}^{m_c} p_{c,k},$$

and normalize to a points-weighted pass-rate in $[0, 1]$,

$$r_c^i = \frac{s_c^i}{S_c} = \frac{\sum_k p_{c,k}\, j_{c,k}^i}{\sum_k p_{c,k}} \in [0, 1].$$

This yields two scalars per rollout, r^i_{PERC} and r^i_{REASON}, which enter the multi-advantage GRPO objective as separate reward components `rubric_perception` and `rubric_reasoning` with equal weights $w_{\text{PERC}} = w_{\text{REASON}} = 0.5$. If a judge call fails to parse, that prompt's rubric components are dropped (soft-fail) and training falls back to the remaining outcome/format advantages, so the reward loop never blocks optimization.

A.3 Benchmarks and Evaluation Protocol

We evaluate along two capability axes—general visual understanding and multimodal math reasoning—and additionally report hallucination diagnosis.

General visual understanding. We use four benchmarks. **MMMU** (Yue et al., 2024) contains ∼11.5K college-level four-choice questions spanning six disciplines, testing cross-modal reasoning and subject knowledge. **MMMU-Pro** (Yue et al., 2025) increases the difficulty with ten choices per question and a vision-only setting in which all text is embedded in the image, requiring robust visual parsing. **RealWorldQA** (xAI, 2024) comprises ∼700 real-world images paired with spatially grounded questions. **VisNumBench** (Weng et al., 2025) targets visual number sense with ∼1.9K questions covering seven numerical attributes and four estimation tasks.

Multimodal math reasoning. We use two benchmarks. **MathVerse** (Zhang et al., 2024) consists of ∼2.6K diagram-centric problems (e.g., geometry, functions), each rendered in multiple visual–text variants to disentangle genuine visual understanding from textual shortcuts; evaluation follows step-by-step chain-of-thought correctness. **MATH-Vision** (Wang et al., 2024b) presents ∼3K competition-grade problems across 16 disciplines and five difficulty levels.

Hallucination diagnosis. HallusionBench (Guan et al., 2024) uses binary yes/no questions with unambiguous ground truth to attribute failures to either language-side hallucination (ignoring visual context) or visual-illusion errors (misinterpreting the image).

Case 1 — Decision-criterion inversion (MMMU)

Food	Number of people who ate that food	Number of people who ate the food and got sick
Cold chicken	86	34
Potato salad	54	38
Egg sandwiches	76	40
Fruit pie and cream	32	12
Cheese	48	12

Question: According to the table below, which food is the most likely cause of the outbreak of food poisoning?
A.Cold chicken B. Potato salad C. Egg sandwiches D. Fruit pie and cream **[GT:B]**

Both models extract the table correctly and compute the *same* attack rates (sick / ate): Cold chicken 0.40, Potato salad 0.70, Egg sandwiches 0.53, Fruit pie 0.38, Cheese 0.25. The divergence is purely in the conclusion.

Rubric-RLVR (✓ → B). "…The highest ratio is for Potato salad (38/54), ≈ 0.70. This indicates that more people who ate potato salad got sick compared to other foods, suggesting it is the most likely cause." → \boxed{B}

Self-Reward (✗ → D). "…The fruit pie and cream has the **lowest** illness rate (0.375) … This suggests that the fruit pie and cream is the food most likely to be responsible for the outbreak, as it has the lowest rate of illness per person who ate it." → \boxed{D}

Takeaway. Identical perception and arithmetic, but the self-reward model **inverts the decision rule** (picks the *lowest* attack rate). Outcome/self-reward rewards the final token, so a model can learn fluent setup without a consistent reasoning policy; the rubric's *reasoning* sub-score penalizes exactly this criterion flip.

Metric and aggregation. We report accuracy on all benchmarks. For multiple-choice questions accuracy is computed by exact option matching; for open-ended questions we follow Li et al. (2025) and use Qwen2.5-14B as a judge to determine answer correctness against the reference. The *general visual understanding* score reported in the main text is the unweighted mean of the four understanding benchmarks, and the *math reasoning* score is the unweighted mean of the two math benchmarks; the *overall* score is the mean of these two category scores. Reporting the two category means separately, rather than a single pooled average, is deliberate: it keeps the perceptual and reasoning effects visible rather than letting one compensate for the other.

B Additional Results

C Case Study

C.1 More Qualitative Examples

Case Studies: OURS RLVR vs. Self-Reward (3B) Qualitative comparison of two Qwen2.5-VL-3B models trained from the same cold-start checkpoint but with different reward signals, evaluated greedily (n=1, T=0) on the same benchmark questions:

Each image below is the exact graded item; answers are the model's own `\boxed{}` extraction.

C.2 Examples of Induced Rubrics

For each prompt, the Rubricator induces a shared two-category rubric: perception over description, reasoning over thinking and answer. The Verifier scores each rollout. Each prompt has its own rubric

Case 2 — Relative-length perception (HallusionBench)

Question. Are the yellow line and the red line the same length in the image? A. Yes B. No.**[GT:B]**

Rubric-RLVR (✓ → B). "…the yellow line … is longer than the red line. The red line appears to be about half the length of the yellow line … The yellow line extends further to the right than the red line, confirming that they are not the same length." → \boxed{B}

Self-Reward (✗ → A). "…Upon visual inspection, the red line and the yellow line appear to be the same length. They both span the same horizontal distance across the image … the lengths … are indeed the same." → \boxed{A}

Takeaway. A direct perceptual measurement. The self-reward model asserts equality against the visual evidence; the rubric model reads the relative lengths correctly. The rubric's *perception* sub-score directly supervises this kind of grounded visual judgment.

Case 3 — Text grounding vs. visual prior (HallusionBench)

Question. According to the text given in the image, is this logo for Red Bull? A. Yes B. No" (Logo = two red bulls + yellow circle, but the caption reads "Red Deer.) **[GT:B]**.

Both models *read* the caption "Red Deer". They differ in what they trust.

Rubric-RLVR (✓ → B). "…the crucial difference is the text at the bottom, which reads **'Red Deer.'** This is not the same as 'Red Bull' … this logo is not for Red Bull but rather for a company named 'Red Deer.'" → \boxed{B}

Self-Reward (✗ → A). "…the text 'Red Deer' is written in red. The bulls are a well-known symbol associated with the Red Bull … brand … The text 'Red Deer' suggests a connection to 'Red Bull' … this logo is indeed associated with Red Bull." → \boxed{A}

Takeaway. The question explicitly says "according to the text." The self-reward model lets its visual/brand prior override the text it just transcribed; the rubric model keeps the answer grounded in the stated evidence. This is the textbook HallusionBench failure mode that the perception rubric is designed to suppress.

Case 4 — Over-reasoning a correct answer away (HallusionBench)

Question. "Are there any carrots in this figure? A. Yes B. No" (A set of tools next to a small carrot illustration.) **[GT:A]**.

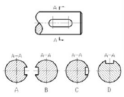
Both models perceive the carrot illustration.

Rubric-RLVR (✓ → A). "…there is an illustration of a carrot on the left side … Since the image includes an illustration of a carrot, it can be concluded that there are indeed carrots present in the image, even though they are not real." → \boxed{A}

Self-Reward (✗ → B). "…the carrot illustration … is not part of the actual collection of tools … The carrot is not physically present in the image with the tools. Therefore, the actual image does not contain any real carrots." → \boxed{B}

Takeaway. The self-reward model invents a spurious real-vs-illustration distinction the question never asked for and reasons itself out of the correct answer. The rubric model answers the question as posed. Process supervision rewards staying on the asked sub-goal rather than manufacturing extra constraints.

Case 5 — Spatial / engineering-drawing reasoning (MMMU)



Question. Select the correct cross-section from the A–A cross-section in the following figure. A / B / C / D (A keyed shaft; the A–A keyway sits on the left.) **[GT:A]**.

Both models agree the cut is a vertical section through the centre; they disagree on which option's keyway position matches.

Rubric-RLVR (✓ → A). "…Option A shows a circular cross-section with a cut on the **left** side. This matches the description of the A-A cut … Option B shows a … cut on the right side … does not match." → \boxed{A}

Self-Reward (✗ → B). "…A. … the cut is not aligned with the centerline … incorrect. B. … the cut is on the right side … aligned with the centerline, which matches the A-A cut." → \boxed{B}

Takeaway. A spatial-correspondence task. The self-reward model invents an "aligned with the centerline" criterion and applies it inconsistently, flipping left↔right; the rubric model matches the keyway position to the drawing. Faithful spatial perception is exactly the perception sub-goal the rubric scores.

Case 6 — Anatomical structure perception (MMMU)

Question. Which organ(s) are abnormally large in this abdominal radiograph? A. Kidneys B. Small Intestines C. Liver D. Spine.**[GT:A]**.

Rubric-RLVR (✓ → A). "…the area where the kidneys would be … appears as a large, rounded structure … the kidneys might be enlarged … The rounded, large structure on either side of the spine is consistent with the appearance of enlarged kidneys." → \boxed{A}

Self-Reward (✗ → C). "…the organs on the right side, particularly the liver, appear to be enlarged … The size of the liver in this image suggests it is larger than normal … the liver is the organ that appears abnormally large." → \boxed{C}

Takeaway. A domain-specific perception task. The self-reward model defaults to the most prototypically large abdominal organ (liver) instead of localising the actual enlarged structure flanking the spine; the rubric model grounds its answer in the bilateral para-spinal masses (the kidneys). The perception sub-goal rewards reading the image over invoking a category prior.

Case 1

Question: A square with side length 1 and an equilateral triangle are arranged as shown. Find the area of triangle ABC.
Ground truth: 1/4

Perception criteria (max 12):
(5 pt) Does the description correctly identify and enumerate all key shapes (square, equilateral triangle, triangle ABC) and their spatial arrangement (adjacency, shared side)?
(4 pt) Does the description specify the relevant labeled vertices (A, B, C) and their positions relative to the shapes?
(2 pt) Is the description self-contained and understandable without access to the image?
(1 pt) Does the description avoid stating or implying the final answer or any calculation/conclusion?

Reasoning criteria (max 13):
(5 pt) Does the reasoning use the visual facts from the description (shapes, side lengths, angles, spatial relations) to construct a solution, rather than guessing or relying solely on language priors?
(4 pt) Are the logical steps valid, with no unsupported leaps or errors in geometric reasoning (e.g., correct use of triangle area formulas, correct calculation of angles or coordinates)?
(4 pt) Is the final boxed answer mathematically correct and equivalent to \frac{1}{4} (accepting any equivalent form)?

Verifier reward (normalized, perception / reasoning). good student = 1.00 / 1.00; weak student = 0.08 / 0.00.

Figure 8: Geometry reasoning rubric for solving the area of a constructed triangle using spatial relations between a square and an equilateral triangle. The rubric evaluates both visual understanding of geometric configuration and correctness of analytical reasoning.

adapted to the task. For more details, please refer to Fig. 8- 10

D Prompts

RL training format prompt.

```
You are tasked with analyzing an
    image/video to generate a detailed
description to help you answer the
    question. First analyze the image
    or video and produce a
self contained description detailed
     enough that can lead to the
    correct answer. Do not
include the final answer in the
    description. Wrap the entire
    description in
<description> </description> tags.
Next, engage in an internal dialogue
    and include self-reflection or
    verification in your
reasoning process. Provide your
    detailed, step-by-step reasoning
    based on the image/video
description information and image/
    video, and enclose this part
    within <think> </think> tags.
Finally, provide a single word or
    phrase answer to the question in \
    boxed{}.
The output format should be: <
    description> image/video
    description here </description>
```

Case 2

Question: Substance X is a concentrated acid that turns sucrose black. The figure shows transformations among A, B, C, D, and X. Which statement is correct?
Ground truth: D

Perception criteria (max 12):
(5 pt) The description correctly identifies and names all key labeled components (X, A, B, D, C) and their spatial arrangement as shown in the image.
(4 pt) The description accurately describes the reaction arrows and their directionality, including the reactants and products for each step.
(2 pt) The description includes relevant text labels (e.g., H_2O, O_2) and notes any partial visibility or omission as seen in the image.
(1 pt) The description is self-contained and understandable without access to the image.

Reasoning criteria (max 15):
(5 pt) The reasoning uses the visual facts from the description (e.g., reaction relationships, labels, and context) to inform chemical identification and analysis, rather than guessing from language priors.
(4 pt) The reasoning proceeds through logically valid steps, with no unsupported leaps or contradictions, and addresses the chemical properties and industrial context relevant to the question.
(4 pt) The final boxed answer is correct (equivalent to D, regardless of surface form).
(2 pt) The reasoning explicitly considers and evaluates all provided answer options (A–D), justifying the selection or rejection of each.

Verifier reward (normalized, perception / reasoning). good student = 1.00 / 1.00; weak student = 0.00 / 0.00.

Figure 9: Chemistry reasoning rubric for identifying reaction relationships and analyzing qualitative chemical behavior based on visual representations of substances and their transformations. The rubric assesses accuracy of visual interpretation and logical chemical inference.

Case 3

Question: Chess Ranger: pieces move as in chess; only capture moves allowed; the king can be captured; goal is one piece remaining. How many steps are needed?
Ground truth: D

Perception criteria (max 12):
(5 pt) The description correctly identifies all chess pieces present, including their types (Queen, Knight, Bishop, Rook) and their exact board positions (e.g., a7, a6, f8, f1).
(4 pt) The description specifies the board layout, including the 8x8 grid, file and rank labels, and the alternating square colors.
(2 pt) The description is self-contained and understandable without viewing the image.
(1 pt) The description does not state or imply the final answer or solution steps.

Reasoning criteria (max 14):
(5 pt) The reasoning uses the visual facts from the description (piece types, positions, board layout) to determine the solution, rather than guessing or relying solely on language priors.
(4 pt) The reasoning provides a logically valid sequence of capture moves, with each move justified by chess rules and the board state.
(3 pt) The reasoning correctly determines the number of steps required to reach a single remaining piece.
(2 pt) The final boxed answer is equivalent to option D (3 steps), regardless of surface form.

Verifier reward (normalized, perception / reasoning). good student = 1.00 / 1.00; weak student = 0.08 / 0.00.

Figure 10: Logical and spatial reasoning rubric for the Chess Ranger puzzle, assessing the ability to interpret board layout, apply chess move rules, and deduce the minimal capture sequence required to reach the goal state.

```
<think> reasoning process here </
    think> \boxed{FINAL ANSWER here}.
```

Rubricator prompt.

```
You are an expert designer of shared
    scoring rubrics for vision-
    language reasoning.
A student model answers an image
    question in three parts: a visual
    DESCRIPTION
(<description>), a step-by-step
    REASONING (<think>), and a final
    boxed ANSWER.
You are given a strong TEACHER
    reference (its caption and its
    chain-of-thought) and a
batch of STUDENT responses. The
    teacher reference is high-quality
    but NOT guaranteed
perfect and must NOT be treated as
    the only valid wording, method, or
     path.

Your job: induce ONE shared rubric
    with EXACTLY TWO fixed categories
        "perception"
and "reasoning"     and, inside each
    category, generate concrete
    criteria distilled from
the teacher-vs-student contrast. The
    same rubric will later be used by
    a verifier to
score every student response
    independently.

# Inputs
[Question]: {question}
[Teacher caption (perception
    reference)]: {teacher_caption}
[Teacher reasoning (reasoning
    reference)]: {teacher_cot}
[Ground-truth final answer]: {
    ground_truth}
[Student descriptions]:
{student_descriptions}
[Student reasonings + answers]:
{student_reasonings}

# Category 1     perception  (
    evaluates the <description> only)
Induce 3  6   criteria about
    description QUALITY AND VISUAL
    GROUNDING. Good criteria check:
- whether specific visual facts
    needed to solve the problem are
    perceived correctly
  (objects, counts, spatial relations
      , labels/text, values read off
      the image);
- whether the description covers the
    key visual information the
    question depends on;
- whether the description is self-
    contained (understandable without
    the image);
```

```
- whether the description AVOIDS
    leaking the final answer or
    conclusion.
Hard rules: each criterion must be
    judgeable True/False from a single
     description alone;
do NOT reward copying the teacher
    caption's wording; do NOT require
    matching the teacher;
a criterion that rewards stating the
    final answer inside the
    description is FORBIDDEN.

# Category 2     reasoning  (
    evaluates the <think> + boxed
    answer)
Induce 3  6  criteria about reasoning
     QUALITY AND CORRECTNESS. Good
    criteria check:
- whether the reasoning actually USES
     the perceived visual facts rather
     than guessing
  from language priors;
- whether the steps are logically
    valid with no unsupported leaps;
- whether the final boxed answer is
    correct (equivalent to {
    ground_truth}; accept any
  surface form / equivalent
      expression).
Hard rules: judgeable True/False from
     a single response; do NOT reward
    matching the
teacher reasoning's wording or method
    ; accept different-but-valid
    methods.

# Point semantics (sharpen, do NOT
    spread points evenly)
5 = decisive bottleneck (failing it
    means the part fundamentally fails
     its job)
4 = strong differentiator
3 = rare intermediate
2 = supporting advantage
1 = low-risk / surface requirement (
    pure-format criteria max out at 1)
Within each category, usually only 1
     2  criteria deserve 4  5  ; the
    top-weighted criteria
should sum to more than half of that
    category's total.

# Output (JSON only, no prose)
{
  "schema_version": "vsr1.
      hybrid_rubric.v1",
  "categories": [
    {"name": "perception",
     "rubrics": [{"criterion_id": "p1
         ", "criterion": "<concrete
         check>", "points": 5}, ...],
     "maximum_score": <sum of
         perception points>},
    {"name": "reasoning",
     "rubrics": [{"criterion_id": "r1
         ", "criterion": "<concrete
         check>", "points": 5}, ...],
     "maximum_score": <sum of
         reasoning points>}
  ]
}

# Constraints
- schema_version must equal "vsr1.
    hybrid_rubric.v1".
- Exactly two categories, named "
    perception" and "reasoning", in
    that order.
- Each category has 3 to 6 criteria;
    criterion_id unique (p1,p2,... /
    r1,r2,...).
- points are integers 1  5  ; each
    category's maximum_score equals
    the sum of its points.
- Prefer fewer, sharply
    discriminative criteria over many
    weak/overlapping ones.
- Output ONLY the JSON object.
```

Verifier prompt.

```
You are a strict grader for vision-
    language reasoning. Given one
    question, a shared
two-category rubric, and a batch of
    student responses (each split into
     a DESCRIPTION
part and a REASONING+ANSWER part),
    score every response independently
    .

[Question]
{question}

[Rubric]
{rubric}

[Responses]
{answers}

# Scoring rules
For each response, go criterion by
    criterion:
- Score the "perception" criteria
    ONLY against that response's
    DESCRIPTION part.
- Score the "reasoning" criteria ONLY
     against that response's REASONING
    +ANSWER part.
- judgement is `true` iff the part
    clearly satisfies the criterion,
    else `false`.
  No partial credit within a
      criterion. If a criterion
      bundles multiple conditions,
  all must hold for `true`.
- A different-but-valid method/
    wording that meets the criterion
    is still `true`.
- Do NOT introduce criteria beyond
    the rubric. Do NOT reward length,
    confidence, or
  resemblance to a reference unless a
       criterion explicitly asks for
```

```
      it.
- Score every response independently;
     do not compare responses to each
    other.
- For each category, `score` = sum of
     that category's criterion points
    where judgement is true.

# Output (JSON only, no prose)
{
  "schema_version": "vsr1.
      hybrid_verifier.v1",
  "answers": [
    {"answer_index": 1,
     "perception": {"judgement": [
         true, false, ...], "score": <
         int>},
     "reasoning":  {"judgement": [
         true, ...],        "score": <
         int>}},
    ...
  ]
}

# Constraints
- schema_version must equal "vsr1.
    hybrid_verifier.v1".
- `answers` covers every input
    response, answer_index 1-based in
    input order.
- Each category's `judgement` length
    equals that category's number of
    criteria, in order.
- `score` equals the sum of points of
     the true criteria in that
    category.
- Output ONLY the JSON object.
```