SPLIT-RL: STAGED PERCEPTION–LANGUAGE REASONING TRAINING WITH CLAIM-LEVEL ADVANTAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision–Language (VL) reasoning requires a model to both extract relevant and accurate information from an image (*visual reasoning*, VR), and to infer the answer from it (*language reasoning*, LR). Reinforcement learning with verifiable rewards typically trains both through a single chain-of-thought with a final-answer reward. This gives every CoT token the same sequence-level advantage, failing to distinguish capability specific errors. We propose **SPLIT-RL**, a staged post-training approach that trains VR and LR in disjoint phases. Because a group's rollouts differ along one capability at a time, the group-relative advantage isolates it, and each phase is optimized using phase-specific reward. We further introduce Claim-Level Advantage (CLA-GRPO), which decomposes VR-phase rollouts into atomic visual claims and provides a fine-grained advantage at claim level based on visual-type group formation. Although trained in two phases, trained policy is evaluated like GRPO model, with a single CoT call at inference time. Under this protocol, SPLIT-RL improves average accuracy over GRPO by 1.4–6.1 points across Qwen3-VL models from 2B to 30B-A3B and InternVL3.5-8B. Evaluating each capability using an oracle based diagnostic shows that answer-only GRPO leaves perception unchanged, whereas SPLIT-RL improves both VR and LR.

1 INTRODUCTION

Vision–Language (VL) reasoning pairs an image with a question that cannot be answered from text knowledge alone. Solving it involves two capabilities: interpreting the question and extracting the visual information it requires - Visual Reasoning (VR), followed by inferring the answer from that information - Language Reasoning (LR). These capabilities underpin reliable multimodal systems such as chart and diagram QA (Masry et al., 2022), robotic manipulation (Brohan et al., 2023) and medical imaging (Li et al., 2025). Modern Vision Language Models (VLMs) (Bai et al., 2025) answer such questions by generating a Chain-of-Thought (CoT) (Wei et al., 2022) followed by a final answer, and are commonly post-trained using reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024).

In this setup, a single CoT stream interleaves VR and LR, which we call the *entangled CoT* and a final-answer correctness reward is optimized using GRPO (DeepSeek-AI, 2025). GRPO improves performance, but the entangled CoT makes credit assignment coarse at two levels. First, because each rollout mixes VR and LR, the group-relative advantage is not capability-specific: it cannot directly optimize either capability in isolation. Second, a rollout may contain multiple visual claims and reasoning steps, some correct and some not, yet GRPO assigns a single sequence-level advantage based on the final-answer reward. As a result, incorrect claims may be reinforced when the final answer is correct, while correct claims may be penalized whenever answer is wrong (Figure 1, top). The entanglement also makes capability-specific rewards hard to define in the first place: with VR and LR interleaved token by token, there is no clean segment on which to place a perception reward.

Recent works (Zhang et al., 2026; Xiao et al., 2026) add perception-specific rewards and training strategies to address this limitation. PEARL (Zhang et al., 2026) builds a perception checklist for verification; Perception-R1 (Xiao et al., 2026) prompts an LLM to generate perception claims and verifies the CoT against them, and Vision-SR1 (Li et al., 2026) re-prompts the policy on its own perception, without the image, to obtain a self-reward. However, these methods either require an

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.

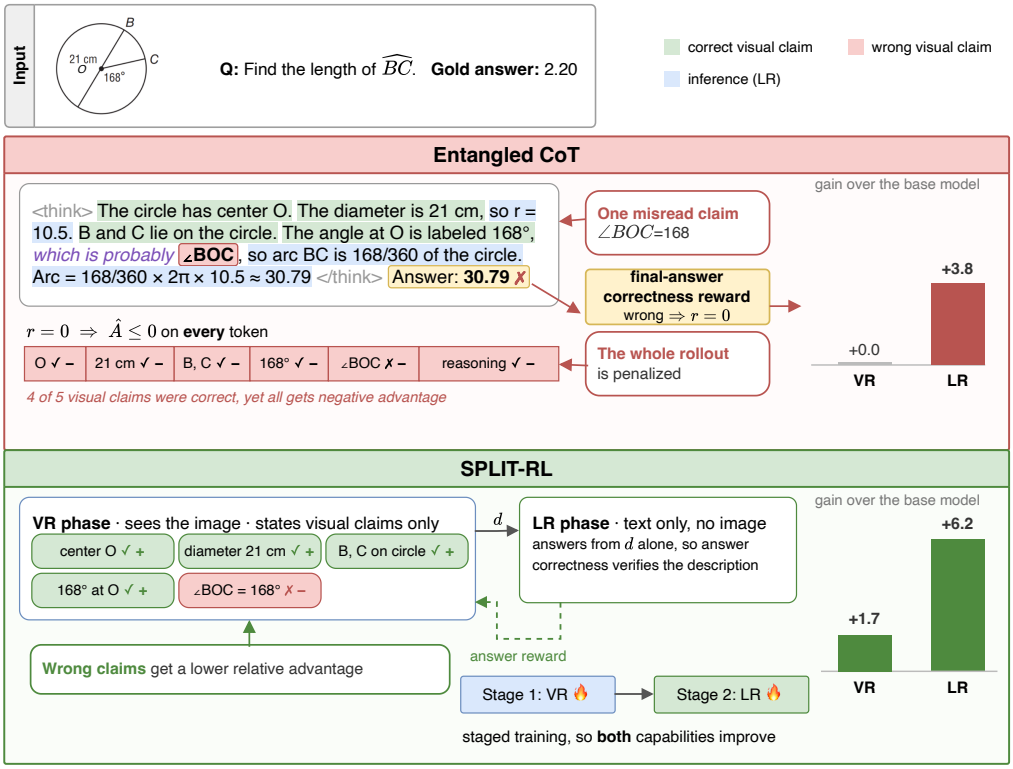


Figure 1: **Entangled CoT vs. SPLIT-RL. Top:** One misread visual claim produces a wrong answer, assigning every token the same nonpositive advantage including four correct claims. With a fixed oracle (Section 4.3.2), answer-only training improves LR (+3.8) but not VR (+0.0). **Bottom:** SPLIT-RL separates visual description from text-only reasoning, using answer correctness to assess description sufficiency. CLA-GRPO assigns claim-level advantages, and training VR before LR improves both capabilities (+1.7 VR, +6.2 LR).

offline data-curation stage or optimize perception jointly with the final answer in a single response, and all retain sequence-level perception rewards.

To this end, we propose **S**taged **P**erception–**L**anguage reason**I**ng **T**raining with **R**einforcement **L**earning (**SPLIT-RL**), which sequentially trains VR and LR as two disjoint stages of a single policy (Figure 1, bottom). In the VR stage, the model sees the image and the query and performs VR to generate query-relevant visual description, with no reasoning and no answer. In the LR stage, the same model receives only this description and the query and reasons to the answer. Because the reasoner never sees the image, the description becomes an information bottleneck: a rollout succeeds only if the description alone suffices to answer, which pushes the model toward more complete and accurate perception without ground-truth descriptions or VR-specific data. Conversely, removing the image forces the reasoner to derive the answer from stated evidence alone, grounding every step in text rather than in shortcuts from the image.

While staged training enables disjoint VR/LR optimization and capability-specific relative advantage computation, credit assignment within the VR description remains coarse. SPLIT-RL therefore introduces Claim-Level Advantage (CLA-GRPO), which brings fine-grained credit assignment to the VR stage. CLA-GRPO decomposes the description into atomic visual claims, verifies each against the image, and groups claims of the same type (e.g., color or size) into sibling groups, so that each claim's advantage is computed relative to claims of the same kind. This provides fine-grained supervision for the correctness of individual visual claims. However, correctness alone does not ensure that the description contains all visual information required to answer the query. We therefore complement VR stage with a completeness reward named recall reward that encourages the VR description to capture all query-relevant evidence needed by the LR stage.

On six visual benchmarks, evaluated under the standard GRPO setting of single stage CoT inference, SPLIT-RL outperforms entangled CoT GRPO across five backbones, by $+1.4$ to $+6.1$ average points, including $+2.7$ on Qwen3-VL-8B. Using our diagnostic probe to measure VR and LR improvement, we show that SPLIT-RL improves LR by $+6.2$ over the base model, $+2.4$ over GRPO, and VR by $+1.7$, whereas GRPO leaves VR unchanged (see figure 1).

In summary, our main contributions are:

- We propose SPLIT-RL, a staged post-training approach that disentangles VR and LR within a single policy and trains them in sequence, using the description as a bottleneck so that answer correctness verifies perception.
- We make VR training capability-specific by rewarding description completeness through a recall reward and correctness through claim-level advantages from CLA-GRPO.
- We show consistent gains over GRPO across five backbones spanning model scales, architectures, and families, with extensive ablations analyzing capability gains.

2 Related Work

Post-training RL for vision–language reasoning RLVR (Lambert et al., 2024) has become the standard recipe for eliciting CoT reasoning from language models. DeepSeek-R1 (DeepSeek-AI, 2025) further made the training efficient by using GRPO. A rapid line of work has since transferred this recipe to VLMs. Initial works such as Vision-R1 (Huang et al., 2025) and R1-OneVision (Yang et al., 2025) pair a multimodal chain-of-thought cold start with GRPO, while MM-Eureka (Meng et al., 2025) and LMM-R1 (Peng et al., 2025) show that rule-based RL alone yields stable gains on multimodal math. OpenVLThinker (Deng et al., 2025b) alternates SFT and RL, and VL-Rethinker (Wang et al., 2025a) and R1-VL (Zhang et al., 2025a) target reflection and step-wise rewards. RLVR has also been specialized to perception tasks such as fine-grained reasoning, grounding, detection and segmentation (Kumar et al., 2026; Shen et al., 2025; Liu et al., 2025), to video (Feng et al., 2025), medical imaging (Lai et al., 2025) and GUI agents (Luo et al., 2025; Lu et al., 2025). Across this breadth the interface is unchanged: a single entangled CoT interleaves perception and reasoning, trained by final-answer correctness alone. This is the setting we revisit in this work.

Capability specific VLM post-training A growing body of work (Wu et al., 2026; Wang et al., 2025e) has shown that VL reasoning interleaves VR and LR capabilities and final answer reward is insufficient. Process reward models (Lightman et al., 2023) score each intermediate step instead of only the outcome and VisualPRM (Wang et al., 2025c), MM-PRM (Du et al., 2025) and VR-PRM (Chen et al., 2025) train multimodal PRMs for vision–language reasoning. However, these methods require step-level annotations to train and are most often inferred for best-of-N selection at inference. Another line of work include curriculum learning where model is trained from from easy to hard data (Deng et al., 2025a) or by perception to reasoning data Wu et al. (2026). These stage data and therefore depend on curated corpora. We instead stage capability optimization, with a verifiable reward attached to each stage. Another family uses tools to revisit the image while reasoning such as sketching on it (Hu et al., 2024), zooming and cropping (Zheng et al., 2025b; Zhang et al., 2025b), running pixel-space operations (Wang et al., 2025b), or invoking external visual tools (Su et al., 2025). These methods improve access to visual evidence, but come at the cost of additional compute (Wang et al., 2026). We keep a pure-text interface and instead improve inherent capability. Our component-level signal is related in spirit to atomic fact verification for text generation (Min et al., 2023). Similar to our work, MoCA (Wang et al., 2026) and Vision-SR1 (Li et al., 2026) use perception as a bottleneck to improve VR capability of the model. MoCA uses an external LLM while Vision-SR1 uses the same model as verifier. However, they still perform joint optimization and their VR specific reward remains coarse at sequence level unlike SPLIT-RL which performs a staged optimization and provides a fine-grained advantage at claim level using CLA-GRPO.

3 SPLIT-RL

We present SPLIT-RL, a staged post-training approach that trains VR and LR within a single policy (Figure 2). We first discuss the disjoint VR and LR formulation(Section 3.1). Then we present our

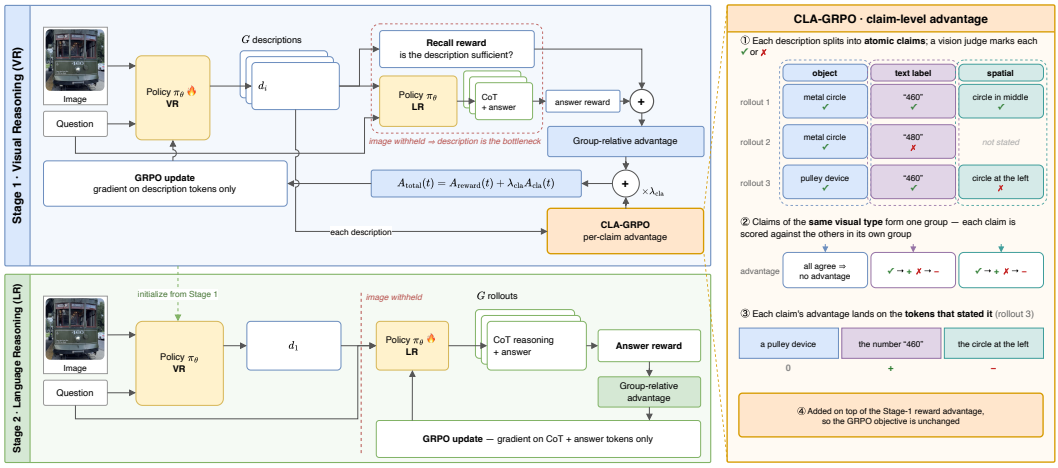


Figure 2: **SPLIT-RL.** One policy is trained in two stages. **Stage 1 (VR):** Samples G descriptions per image–question pair and updates description tokens using recall, answer-correctness, and claim-level signals. **Stage 2 (LR):** Starting from Stage 1, generates one description d and samples G reasoning–answer rollouts from (Q, d) without the image; GRPO updates only reasoning and answer tokens. **Right:** CLA-GRPO verifies atomic claims against the image, groups them by visual type across rollouts, and assigns claim-level advantages to their tokens.

staged training approach (Section 3.2). Finally, we introduce VR- specific signals, CLA-GRPO and the recall reward, in Section (3.3) CLA-GRPO and recall reward.

3.1 Disjoint visual and language reasoning

Let I be an image, Q a question and $a^\star$ the gold answer and π_θ is the policy itself. Standard GRPO (Shao et al., 2024) samples a group of G rollouts from the policy, $\{o_1, \ldots, o_G\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid I, Q)$, scores each one by final-answer correctness $r_i = r(a^\star, o_i) \in \{0, 1\}$, and computes the advantage from the group statistics,

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}, \tag{1}$$

which is then optimized with the clipped surrogate objective 11. The same scalar $\hat{A}_i$ is broadcast uniformly over all tokens of o_i, so tokens that perceive and tokens that reason receive identical credit, and the reward cannot tell which capability failed.

In SPLIT-RL, we split a single generation into two phases performed by the same policy π_θ. In the VR phase, π_θ acts as a *describer*: given the image I and question Q, it produces a description d of the visual content relevant to Q, with no reasoning and answer. In the LR phase, the same π_θ acts as a *reasoner*: it receives only Q and d (no image) and derives the answer $\hat{a}$, leading to equation 2

$$d \sim \pi_\theta(\cdot \mid I, Q), \qquad \hat{a} \sim \pi_\theta(\cdot \mid Q, d), \qquad r_{\text{ans}}(d) = r(a^\star, \hat{a}) \in \{0, 1\}. \tag{2}$$

Description as a bottleneck. Because the reasoner is conditioned on d alone, any visual fact that d omits—or states incorrectly—cannot be recovered downstream. The answer reward $r_{\text{ans}}(d)$ therefore measures whether d is *sufficient* for Q: a verifiable signal for perception that needs no caption-level supervision, turning every $(I, Q, a^\star)$ triple into VR training data. Crucially, the reasoner that supplies this signal *is* π_θ itself. The verifier costs no extra parameters, and it grades each description against what the model can actually do with it, and the sufficiency bar is set by the model's own reasoning rather than an external judge.

3.2 Staged training with capability-specific rollouts

The factorization in equation 2 lets us optimize VR and LR in sequence rather than jointly. Both stages update the same parameters θ with the objective equation 11; they differ in what is sampled, what is rewarded, and which tokens receive the gradient.

Stage 1: VR. For each (I, Q), the policy samples G descriptions $(d_1, \ldots, d_G) \sim \pi_\theta(\cdot \mid I, Q)$, and each is scored by the same policy reasoner, $\hat{a}_i \sim \pi_\theta(\cdot \mid Q, d_i)$, to obtain the bottleneck answer reward $r_{\text{ans},i}$. The advantage is computed over G sibling descriptions and applied only to description tokens; the answer tokens act as part of the verifier and receive no gradient. Section 3.3 adds two further signals to this stage.

Stage 2: LR. Stage 2 is initialized from the Stage-1 policy. The trained describer produces one on-policy description $d \sim \pi_\theta(\cdot \mid I, Q)$, and the policy samples G answers conditioned on it, $(\hat{a}_1, \ldots, \hat{a}_G) \sim \pi_\theta(\cdot \mid Q, d)$, each rewarded by $r(a^\star, \hat{a}_i)$. The advantage is applied only to reasoning and answer tokens.

Staging gives each capability a learning signal of its own. First, rollouts differ only in the capability under training: Stage-1 rollouts share the reasoner and differ only in what was described, while Stage-2 rollouts share the description and differ only in how it was reasoned over, so the group-relative baseline of equation 1 isolates one capability instead of averaging over both. Second, sequencing lets each stage act on the other's improved behavior: Stage 2 trains the reasoner on the descriptions the tuned describer actually produces.

3.3 Completeness and correctness signals for the VR stage

The bottleneck reward is coarse and indirect: it is a single scalar per rollout, and as noted in Section 3.1, it is set by the reasoner: a sufficient description can still draw a wrong answer, so a low reward caused by a reasoning failure may therefore be mistaken for a perception failure. We therefore add signals computed directly from the description. A description can fail in two ways: it can *omit* a visual fact the question needs, or *misstate* one. We address each with its own signal, a recall reward for completeness (Section 3.3.1) and a claim-level advantage for correctness (Section 3.3.2).

3.3.1 Completeness: recall reward

Completeness is a property of the description, not of the answer, so we score it with a judge that sees the description but never the image. A text-only judge $\mathcal{J}_{\text{recall}}$ is provided (Q, d), and reports whether d alone states every visual fact needed to answer Q,

$$r_{\text{recall}} = \mathcal{J}_{\text{recall}}(Q, d) \in \{0, 1\}. \tag{3}$$

Withholding the image is what makes this a completeness signal: the judge can call d sufficient only if the description itself carries the evidence, so a description that omits a needed fact cannot be rescued by the judge inspecting the image. The overall VR-stage reward is then

$$r_{\text{overall}} = \lambda_{\text{recall}}\, r_{\text{recall}} + \lambda_{\text{ans}}\, r_{\text{ans}}, \tag{4}$$

with r_{ans} as the bottleneck reward of equation 2.

We keep both because they judge sufficiency differently. r_{recall} is direct and unconfounded by reasoning, but a *proxy*: the judge never sees $a^\star$ and scores d against its own notion of what is needed. r_{ans} is *grounded*—anchored to $a^\star$ and to the same π_θ that consumes d—so it optimizes the describer for the model's own reasoning rather than the judge's. Its cost is the confound above: a right description with faulty reasoning scores 0. The two are therefore paired, not substituted.

The recall judge also keeps the VR phase in its role: a description that states or derives the answer is scored $r_{\text{recall}} = 0$ even when sufficient. This closes an obvious shortcut—writing the answer into d would otherwise make r_{ans} trivially high without any real perception—and keeps the bottleneck intact. In Appendix Figure 6 shows an example.

3.3.2 Correctness: claim-level advantage

Both rewards in equation 4 are *holistic*: each returns a single scalar for the whole description, so GRPO assigns the rollout one advantage and credits every claim's tokens identically (Figure 1, top). A description, though, is many claims, each right or wrong on its own. When one misstated claim sits among many correct ones, a single shared advantage cannot push it down without pushing the correct claims down with it. CLA-GRPO instead verifies each claim and gives it its own advantage, acting directly on the tokens that stated it.

Table 1: **Main results.** Accuracy (%) on six benchmarks. Δ rows show percentage-point gains over GRPO for the same backbone. Bold marks the best accuracy per column within each backbone.

Backbone	Method	MVista	MVision	MVerse	WeMath	LogicVista	DynaMath	**AVG**
Qwen3-VL 2B	Base	60.3	18.8	38.8	56.5	34.2	55.9	44.1
	GRPO	60.6	35.9	44.4	54.1	36.7	56.5	48.0
	SPLIT-RL	**65.4**	**42.8**	**46.7**	**62.1**	**44.1**	**63.8**	**54.1**
	Δ vs. GRPO	+4.8	+6.9	+2.3	+8.0	+7.4	+7.3	+6.1
Qwen3-VL 4B	Base	77.1	59.2	61.7	75.6	**63.1**	73.4	68.4
	GRPO	78.0	61.8	66.4	79.8	56.4	74.7	69.5
	SPLIT-RL	**78.7**	**68.4**	**69.3**	**84.0**	62.6	**76.1**	**73.2**
	Δ vs. GRPO	+0.7	+6.6	+2.9	+4.2	+6.2	+1.4	+3.7
Qwen3-VL 8B	Base	77.2	63.2	63.8	79.9	60.4	74.4	69.8
	GRPO	77.9	64.5	67.8	84.6	65.6	75.3	72.6
	VLM-CapCurr (Wu et al., 2026)	**80.0**	44.7	67.6	81.2	60.9	74.6	68.2
	PEARL (Zhang et al., 2026)	77.9	58.2	68.9	84.8	61.5	72.8	70.7
	SPLIT-RL	79.8	**69.4**	**69.8**	**87.4**	**66.0**	**79.1**	**75.3**
	Δ vs. GRPO	+1.9	+4.9	+2.0	+2.8	+0.4	+3.8	+2.7
Qwen3-VL 30B-A3B	Base	79.6	59.2	65.7	75.5	60.4	78.1	69.8
	GRPO	**80.8**	64.9	**74.7**	76.8	64.1	75.1	72.7
	SPLIT-RL	80.0	**66.4**	73.0	**80.3**	**64.2**	**80.9**	**74.1**
	Δ vs. GRPO	−0.8	+1.5	−1.7	+3.5	+0.1	+5.8	+1.4
InternVL3.5 8B	Base	79.2	55.9	60.9	83.9	57.3	70.5	68.0
	GRPO	79.4	62.2	63.2	81.6	57.3	73.6	69.5
	SPLIT-RL	**80.6**	**63.5**	**64.0**	**84.6**	**60.1**	**77.1**	**71.6**
	Δ vs. GRPO	+1.2	+1.3	+0.8	+3.0	+2.8	+3.5	+2.1

Decomposition and verification. Each sampled description d_i is segmented into an ordered sequence of atomic claims, with the character span of each retained,

$$d_i \longrightarrow (c_{i1}, c_{i2}, \ldots, c_{in_i}), \qquad c_{ij} \text{ spanning characters } [\alpha_{ij}, \beta_{ij}). \tag{5}$$

A vision judge $\mathcal{J}_{\text{cla}}$ labels every claim with a correctness, a relevance and a visual-type verdict,

$$\mathcal{J}_{\text{cla}}(I, Q, c_{ij}) = (y_{ij},\, \ell_{ij},\, \tau_{ij}), \qquad y_{ij} \in \{0, 1\}, \quad \ell_{ij} \in \{\text{LOW}, \text{HIGH}\}, \tag{6}$$

where y_{ij} says whether the claim is faithful to the image, ℓ_{ij} its relevance for answering Q, and $\tau_{ij} \in \{\text{object presence, count, color/material, spatial position, text label, geometric relation, } \ldots\}$ names the kind of visual property being claimed. The full judge prompt is given in Appendix 9.7.

Claim-level advantage. A rollout now carries a *set* of verdicts, for which the group-relative estimator equation 1 is not defined. CLA-GRPO treats each visual type τ as one group, so each claim is scored against what the sibling rollouts said about *the same kind of visual property on the same image*. Concretely, the HIGH-relevance claims of type τ made by the G sibling rollouts of a prompt form one sibling group,

$$\mathcal{G}^{\tau} = \{\, (i, j) \;:\; \tau_{ij} = \tau,\;\; \ell_{ij} = \text{HIGH} \,\}, \tag{7}$$

and each claim's own correctness y_{ij} is its reward. Applying the group-relative estimator of equation 1 *within a type* gives a claim-level advantage

$$\hat{A}_{ij} = \begin{cases} \dfrac{y_{ij} - \mu^{\tau_{ij}}}{\sigma^{\tau_{ij}}}, & (i,j) \in \mathcal{G}^{\tau_{ij}} \text{ and } \sigma^{\tau_{ij}} > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad \begin{aligned} \mu^{\tau} &= \text{mean}_{(i,j)\in\mathcal{G}^{\tau}}\, y_{ij}, \\ \sigma^{\tau} &= \text{std}_{(i,j)\in\mathcal{G}^{\tau}}\, y_{ij}. \end{aligned} \tag{8}$$

Credit is thus assigned per claim, not per rollout. Low-relevance claims, and types with a single claim or no disagreement among the siblings, receive zero advantage rather than an undefined normalization.

From claim verdicts to token advantages. To act on the tokens that made a claim, each claim's character span is mapped to the response-token positions it covers,

$$\mathcal{T}_{ij} = \{\, t \;:\; s_t \geq \alpha_{ij} \;\wedge\; e_t \leq \beta_{ij} \,\}, \tag{9}$$

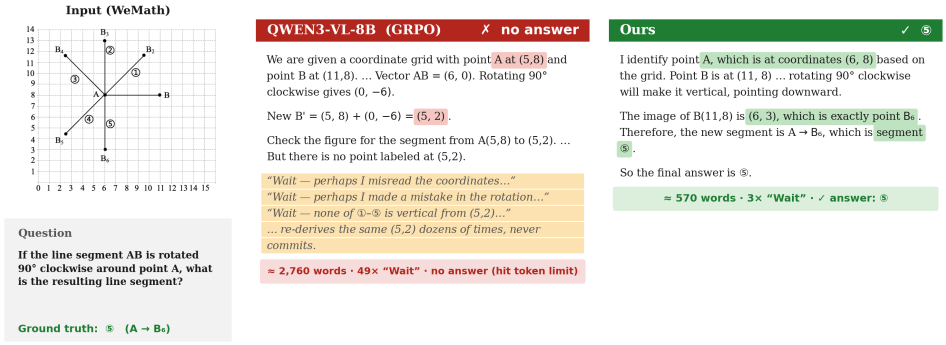


Figure 3: **Qualitative example (WeMath).** Rotating AB by $90°$. GRPO misreads A as $(5, 8)$ and hits the token limit without an answer; SPLIT-RL reads $A = (6, 8)$ and answers correctly.

where $[\alpha_{ij}, \beta_{ij})$ is the character span of claim c_{ij} and $[s_t, e_t)$ the character range of token t, both measured in the decoded response; $\mathcal{T}_{ij}$ collects the tokens lying entirely within the claim. Each claim's advantage is scattered onto its own positions and *added* to the standard reward advantage:

$$A_{\text{total}}(t) \;=\; A_{\text{reward}}(t) \;+\; \lambda_{\text{cla}}\, A_{\text{cla}}(t), \qquad A_{\text{cla}}(t) = \hat{A}_{ij} \;\text{ for the claim } c_{ij} \text{ with } t \in \mathcal{T}_{ij}. \tag{10}$$

Here A_{reward} is the group-relative advantage of r_{overall} (equation 1) broadcast over description tokens, and $A_{\text{cla}}(t) = 0$ for tokens outside any HIGH-relevance claim.

Equation 10 changes nothing about the optimizer: CLA-GRPO is an *additive, already group-normalized advantage channel* that leaves the objective equation 11 unchanged with $\hat{A}_i$ replaced by A_{total} , and reduces to vanilla GRPO when $\lambda_{\text{cla}} = 0$. What it changes is the *granularity* and *targeting* of the learning signal: the holistic rewards give one number to the whole rollout, whereas CLA-GRPO delivers a separately baselined number to the tokens that stated each claim.

4 Experiments

4.1 Experimental Setup

Baselines and implementation details. We compare SPLIT-RL against (i) the zero-shot base model; (ii) a GRPO baseline trained under the same setting as ours; (iii) VLM-CapCurr (Wu et al., 2026) and (iv) PEARL (Zhang et al., 2026). We conduct experiments across model sizes and backbones, using the Qwen3-VL family (Bai et al., 2025) at 2B, 4B and 8B (dense) and 30B-A3B (MoE), and InternVL3.5-8B (Wang et al., 2025d). In SPLIT-RL, a single set of weights serves as both the describer and the reasoner. The GRPO baseline uses 4096 max output tokens, and each SPLIT-RL stage uses 2048. To match the total number of training steps, we train the GRPO baseline for 4 epochs and each SPLIT-RL stage for 2 epochs. All our models are trained on 2 nodes of 8x NVIDIA B200 with 180GB memory. Additional Training data and hyperparameters are detailed in Appendix 9.2.

Benchmarks and evaluation. We evaluate on six standard benchmarks for visual math and logic reasoning benchmarks: MathVista (Lu et al., 2024), MathVision (Wang et al., 2024), MathVerse (Zhang et al., 2024), WeMath (Qiao et al., 2024), LogicVista (Xiao et al., 2024) and DynaMath (Zou et al., 2024). Accuracy is our primary metric; ablations additionally measure VR and LR capability separately (Section 4.3.2).

Inference: SPLIT-RL uses VR–LR split only during training. At inference, the policy receives the image and question and answers in a single CoT call, with the same prompt and decoding settings as the GRPO baseline (more details in Appendix 9.2), so SPLIT-RL adds no inference cost. Notably, GRPO is optimized directly in this single-pass format, whereas SPLIT-RL never is. The policy carries what the stages train, accurate perception and evidence-grounded reasoning, into a format it was never optimized in. Following prior works Wu et al. (2026); Zhang et al. (2026) we perform greedy decoding across all our experiments.

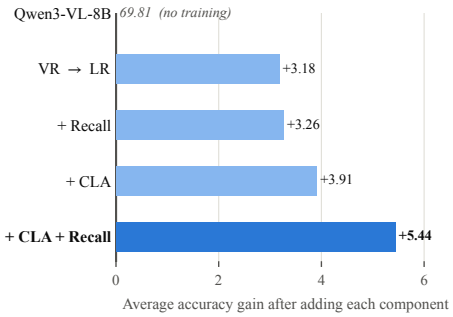


Figure 4: **Contribution of each component**. Per-benchmark results are in Appendix Table 5.

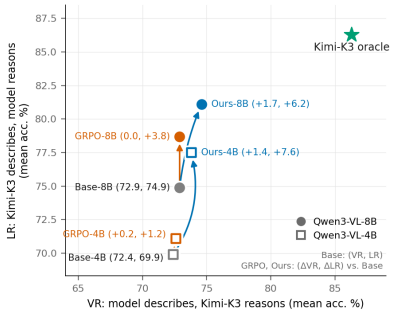


Figure 5: **Diagnostic probe:** each capability in isolation with an oracle.

4.2 Results

4.2.1 Main Results

Table 1 reports accuracy on six benchmarks for five backbones spanning four scales (2B–30B), dense and Mixture-of-Experts architectures, and two model families. SPLIT-RL improves over the GRPO baseline on every backbone, by $+1.5$ to $+6.1$ average points. On the dense Qwen3-VL models, SPLIT-RL improves over GRPO by $+\mathbf{6.1}$ (2B), $+\mathbf{3.7}$ (4B) and $+\mathbf{2.7}$ (8B) points, winning on all six benchmarks at every size. On Qwen3-VL-2B, it gains $+10.0$ points over the base model, against GRPO's $+3.9$. The margin over GRPO narrows as the backbone grows, consistent with stronger base models leaving less headroom on these benchmarks under the same training data and budget; though it remains positive at every scale.

On the Mixture-of-Experts Qwen3-VL-30B-A3B, SPLIT-RL improves over GRPO by $+\mathbf{1.4}$ points, with the largest gains on WeMath, LogicVista and DynaMath ($+3.2$ to $+4.2$), and trails GRPO only on MathVista and MathVision. On InternVL3.5-8B (Wang et al., 2025d), SPLIT-RL improves over GRPO by $+\mathbf{2.1}$ points and on all six benchmarks, showing that the benefits are not specific to one family. Notably, these improvements are obtained without the two-stage interface used during training. For completeness, we also report two-stage inference results in appendix 7.

On Qwen3-VL-8B, SPLIT-RL reaches 75.3 average accuracy, outperforming PEARL (Zhang et al., 2026), which adds a perception-checklist reward to an entangled CoT, by $+\mathbf{4.6}$ points, and VLM-CapCurr (Wu et al., 2026), which stages post-training on curated perception data, by $+\mathbf{7.1}$. [1]

4.2.2 Qualitative results

Figure 3 compares GRPO and SPLIT-RL on a WeMath example (more is in Appendix 9.8). The baseline's error is perceptual, not logical: after one misread coordinate, every later step is consistent but wrong, so no rotation matches an option. Instead of re-reading the figure, it re-derives the same point (49 "Wait"s) until it runs out of tokens, whereas SPLIT-RL needs only three. This matches Section 4.3.2: answer-only training does not correct misread visual facts, and when the evidence is wrong, more reasoning becomes hedging rather than progress.

4.3 Ablation studies

4.3.1 Contribution of each component

Figure 4, isolates the contribution of each component of SPLIT-RL on Qwen3-VL-8B. We start from staged training with only the bottleneck answer reward in the VR stage (VR→LR) and add the recall reward, CLA-GRPO, and both. The LR stage is identical across all variants.

Staging alone already helps. VR→LR uses no supervision beyond answer correctness, exactly like GRPO, yet improves over the base model by $+3.2$ points and slightly exceeds entangled-CoT

[1] Both are evaluated on their released Qwen3-VL-8B checkpoints.

GRPO (72.6). The benefit is clearer when each capability is measured on its own: staging alone already improves the model both as a describer and as a reasoner (Section 4.3.2; Appendix Table 6).

Recall and CLA-GRPO are complementary. Added separately, recall and CLA-GRPO raise the average by only $+0.1$ and $+0.7$ over VR→LR. Combined, they give the best average, 75.3: a $+2.3$ gain over VR→LR, larger than the sum of the two ($+0.8$). Completeness and correctness are therefore distinct requirements on a description, and accurate VR needs both; the superadditive gain indicates the two signals reinforce rather than duplicate each other.

4.3.2 Improvement of each capability

We introduce a diagnostic probe that measures each capability in isolation (Figure 5). For this analysis only, we split generation into the two roles and give the other role to Kimi-K3 (Team et al., 2026) as a fixed oracle. To measure VR, the model describes and Kimi-K3 reasons; to measure LR, Kimi-K3 describes and the model reasons from from the Kimi description alone.

GRPO improves only LR. On Qwen3-VL-8B, the descriptions of the GRPO model have the base accuracy (72.9), while its reasoning improves by $+3.8$. Qwen3-VL-4B shows the same pattern ($+0.2$ VR, $+1.2$ LR) indicating Answer-only RL using entangled CoT leaves perception unchanged.

SPLIT-RL improves both. As a describer, our 8B model raises Kimi-K3's accuracy by $+1.7$ over both the base and GRPO models. As a reasoner, it gains $+6.2$ over the base, $+2.4$ more than GRPO. The per-benchmark results (Appendix Table 6) show that most of this LR gain is already obtained by staging alone (VR→LR). Although LR uses standard GRPO, the model never sees the image and must reason from a visual description only, so every step has to be grounded in stated evidence rather than in shortcuts from the image. Recall and CLA-GRPO increase the LR gain further, even though both act only on the VR stage, because the LR stage is subsequently trained on more complete and correct on-policy descriptions. Better perception and better-grounded reasoning thus reinforce each other as seen in gains of Section 4.3.1. At 4B, SPLIT-RL gains $+1.4$ in VR and $+7.6$ in LR.

4.3.3 Order of staged training

Our recipe trains VR first and LR second. Table 2 reverses this order while keeping the rewards and training budget unchanged. The reversed order (LR→VR) reaches 72.1, only $+2.3$ over the untrained model and 0.5 below answer-only GRPO, whereas VR→LR reaches 75.3 ($+5.5$). Training the reasoner first therefore gives up most of the benefit of staging. The drop concentrates on the benchmarks that depend most on perception, MathVision (-8.2) and WeMath (-5.1), and the reversed order is ahead only on MathVista, by 0.2. A reasoner trained on the descriptions of an untrained describer learns to compensate for incomplete evidence, and the later VR stage cannot undo that. Perception should therefore be established before reasoning is refined.

Stage 1	Stage 2	Mean acc.	Gain
Qwen3-VL-8B		69.8	–
Qwen3-VL-8B (GRPO)		72.6	+2.8
LR	VR	72.1	+2.3
VR	**LR**	**75.3**	**+5.5**

Table 2: Order of staged training. Gains are relative to the base model.

5 Conclusion

We introduced SPLIT-RL, a staged post-training approach that optimizes VR and LR in stages within a single policy. SPLIT-RL uses the description as an information bottleneck, and harnesses final-answer correctness as a supervisory signal for perception, requiring no caption-level labels. The separation further lets each capability be trained with its own reward, and CLA-GRPO turns per-claim verification into a group-relative advantage per visual type that plugs into standard GRPO. On six visual math and logic benchmarks, SPLIT-RL improves over GRPO across model sizes and backbones with average accuracy gains of 1.4–6.1 %. Our ablations show that staging alone already beats entangled-CoT GRPO, that completeness and correctness are distinct requirements whose signals (the recall reward and CLA-GRPO) are complementary, and that order matters – training LR before VR forfeits the benefit of staging, so perception should be established before reasoning is refined. Limitations and future work are discussed in Appendix 9.9.

6 AI USE STATEMENT

Generative AI tools were employed in the course of this work to support multiple activities including writing code for our methods, editing the manuscript's prose and finding existing related works. We additionally relied on such tools to generate and adjust scientific figures. All AI-assisted output was examined by the authors, and we independently confirmed the correctness of the implementation. These tools were not used for intellectual discussion, scientific ideation, or the development of the core research ideas; the conceptual contributions and methodological ideas presented in this work were developed by the authors.

7 REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed training procedure, reward design, and optimization objectives in the main paper. The appendix provides additional implementation details, including training configurations, hyperparameters, data and benchmark descriptions, evaluation protocols, detailed prompts and further experimental results. We use publicly available model backbones and evaluation benchmarks and report the settings necessary to reproduce our experiments.

8 ETHICS STATEMENT

This work develops methods for improving visual and language reasoning in vision–language models. Our experiments use publicly available model backbones and established research benchmarks and do not involve human subjects or the collection of sensitive or personally identifiable information. As with improvements to general-purpose vision–language reasoning, the proposed methods could be incorporated into systems deployed in applications beyond those studied here, including settings in which incorrect model outputs may have real-world consequences. Our experiments are limited to controlled benchmark evaluations and do not establish suitability for such high-stakes deployment.

REFERENCES

Anthropic. Claude sonnet 4.6. Large language model, 2026. URL `https://claude.ai/`. Accessed: 2026-09-18.

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P. Xing, and Liang Lin. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics (ACL)*, 2021.

Xinquan Chen, Chongying Yue, Bangwei Liu, Xuhong Wang, Yingchun Wang, and Chaochao Lu. VRPRM: Process reward modeling via visual reasoning. *arXiv preprint arXiv:2508.03556*, 2025.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*, 2025a.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. OpenVLThinker: Complex vision-language reasoning via iterative SFT-RL cycles. *arXiv preprint arXiv:2503.17352*, 2025b.

Lingxiao Du, Fanqing Meng, Zongkai Liu, Zhixiang Zhou, Ping Luo, Qiaosheng Zhang, and Wenqi Shao. MM-PRM: Enhancing multimodal mathematical reasoning with scalable step-level supervision. *arXiv preprint arXiv:2505.13427*, 2025.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *arXiv preprint arXiv:2503.21776*, 2025.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Raja Kumar, Arka Sadhu, and Ram Nevatia. Dive-k: Differential visual reasoning for fine-grained image recognition. In *International Conference on Learning Representations*, volume 2026, pp. 106415–106433, 2026.

Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, Yuheng Li, Konstantinos Psounis, and Xiaofeng Yang. Med-R1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Xiang Li, Like Li, Yuchen Jiang, Hao Wang, Xinyu Qiao, Ting Feng, Hao Luo, and Yong Zhao. Vision-language models in medical image analysis: From simple fusion to general large models. *Information Fusion*, 118:102995, 2025.

Zongxia Li, Wenhao Yu, Chengsong Huang, Zhenwen Liang, Rui Liu, Fuxiao Liu, Jingxi Che, Dian Yu, Jordan Boyd-Graber, Haitao Mi, and Dong Yu. Vision-SR1: Self-rewarding vision-language model via reasoning decomposition and multi-reward policy optimization. In *International Conference on Learning Representations (ICLR)*, 2026.

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.

Adam Dahlgren Lindström and Savitha Sam Abraham. CLEVR-Math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*, 2022.

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

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. UI-R1: Enhancing efficient action prediction of GUI agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025.

Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. GUI-R1: A generalist R1-style vision-language action model for GUI agents. *arXiv preprint arXiv:2504.10458*, 2025.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of connected and contrasting images. *arXiv preprint arXiv:2404.19753*, 2024.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. LMM-R1: Empowering 3B LMMs with strong reasoning abilities through two-stage rule-based RL. *arXiv preprint arXiv:2503.07536*, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

Nipun Sadvilkar and Mark Neumann. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pp. 110–114, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.nlposs-1.15.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. VLM-R1: A stable and generalizable R1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and Yu Cheng. OpenThinkIMG: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. VL-Rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

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

Haozhe Wang, Qixin Xu, Changpeng Wang, Taofeng Xue, Chong Peng, Wenhu Chen, and Fangzhen Lin. Bad seeing or bad thinking? rewarding perception for multimodal reasoning. In *International Conference on Machine Learning (ICML)*, 2026.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. VisualPRM: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025c.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025d.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. Perception-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2507.06448*, 2025e.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Juncheng Wu, Hardy Chen, Haoqin Tu, Xianfeng Tang, Freda Shi, Hui Liu, Hanqing Lu, Cihang Xie, and Yuyin Zhou. From seeing to thinking: Decoupling perception and reasoning improves post-training of vision-language models. In *International Conference on Machine Learning (ICML)*, 2026. VLM-CapCurriculum.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. Perception-r1: Advancing multimodal reasoning capabilities of mllms via visual perception reward. In *International Conference on Learning Representations*, volume 2026, pp. 26868–26898, 2026.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-Onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Chi Zhang, Haibo Qiu, Qiming Zhang, Yufei Xu, Zhixiong Zeng, Siqi Yang, Peng Shi, Lin Ma, and Jing Zhang. Perceptual-evidence anchored reinforced learning for multimodal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41111–41120, 2026.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-VL: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025a.

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

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient VLMs. *arXiv preprint arXiv:2505.15436*, 2025b.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, Yuwen Xiong, and Richong Zhang. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`, 2025a.

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

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. Math-PUMA: Progressive upward multimodal alignment to enhance mathematical reasoning. *arXiv preprint arXiv:2408.08640*, 2024.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. DynaMath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*, 2024.

9 APPENDIX

9.1 OBJECTIVE FUNCTION

$$J(\theta) = \mathbb{E}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t\in o_i}\min\left(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t}, 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}})\hat{A}_i\right)\Big], \tag{11}$$

where $\rho_{i,t} = \pi_\theta(o_{i,t})/\pi_{\theta_{\text{old}}}(o_{i,t})$

9.2 IMPLEMENTATION DETAILS

Training data: For training, We use the data collected in Wu et al. (2026). The dataset consists of five sources for a total of 19,555 examples (see 3): Math-PUMA (Zhuang et al., 2024), GeoQA170K (Chen et al., 2021), CLEVR-Math (Lindström & Abraham, 2022; Johnson et al., 2017), ArxivQA (Li et al., 2024), and DOCCI (Onoe et al., 2024). Following Kumar et al. (2026), we perform a difficulty signal filtering using 16 rollouts from the base model and computing pass_rate@16 $\in [0, 1]$; examples that are already solved (pass_rate@16 $= 1$) are filtered out, leaving 13,862 training samples. Table 3 shows training data stats.

Table 3: Training data mixture. Difficulty filtering (pass_rate@16=1 removed) reduces 16,195 $\rightarrow$ 13,862 samples.

Source	# examples
Math-PUMA (synthesis)	6,696
GeoQA170K	6,499
CLEVR-Math	2,000
ArxivQA (2$\times$ downsampled)	1,000
DOCCI (2$\times$ downsampled)	3,360
Total	19,555
After difficulty filtering	13,862

Training details: We use EasyR1 (Zheng et al., 2025a) framework for all our training. The vision tower is frozen throughout; only the language model is updated. Images are bounded to [262,144, 1,048,576] pixels. We use a global batch size of 256 and 8 rollouts per sample using temperature of 1.0. With 13,862 training examples this gives 54 optimizer steps per epoch; each SPLIT-RL phase runs 2 epochs (108 steps), and the GRPO baseline runs 4 epochs (216 steps). For the VR stage, we set $\lambda_{\text{recall}} = 0.2$ and $\lambda_{\text{ans}} = 0.8$ in equation 4, and $\lambda_{\text{cla}} = 0.5$ in equation 10; the LR stage follows the standard GRPO setting. We use Claude Sonnet 4.6 (Anthropic, 2026) for all judge calls. We use an LLM judge for the final-answer correctness check, as symbolic verification is not robust Wang et al. (2026). Based on DAPO (Yu et al., 2025), we disable KL penalty and do asymmetric clipping with $\varepsilon_{\text{low}} = 0.2$, $\varepsilon_{\text{high}} = 0.28$. These hyperparameter are consistent across all our experiments including GRPO baseline. For CLA-GRPO decomposition we use sentence level decomposition with PySBD (Sadvilkar & Neumann, 2020), and each component's character span is mapped to response tokens through the tokenizer's offset mapping, keeping only tokens fully contained in the span. A maximum of 60 claims are used for CLA-GRPO and other receive a $A_{cla} = 0$ to penalize long generations.

Inference setup All models are evaluated under the same budget: a maximum of 16,384 new tokens and greedy decoding, with one generation per question. All inference including SPLIT-RL are done using a standard CoT followed by a final answer, with the inference time prompt shown below. Rather than matching answer strings symbolically, we score a response with an LLM judge that is shown the gold answer and the model's full generation, and decides whether the generation reaches the correct answer.

Table 4: **Order of staged training** on Qwen3-VL-8B, per benchmark; the summary is Table 2. Both orders use the same VR-stage rewards (CLA + recall). Δ is the change in average accuracy relative to GRPO. Best per column in **bold**.

Method	MVista	MVision	MVerse	WeMath	LogicVista	DynaMath	**AVG**	Δ
Qwen3-VL-8B (GRPO)	77.9	64.5	67.8	84.6	65.6	75.3	72.6	—
LR → VR	**80.0**	61.2	68.8	82.3	62.4	77.7	72.1	−0.5
VR → LR (SPLIT-RL)	79.8	**69.4**	**69.8**	**87.4**	**66.0**	**79.1**	**75.3**	**+2.7**

Inference prompt

```
<question>

You FIRST think about the reasoning process as an internal monologue and then provide
     the final answer. The reasoning process MUST BE enclosed within <reason> </reason>
      tags. The final answer MUST BE put in \boxed{} (for a multiple-choice question,
     put only the option letter inside \boxed{}).
```

9.3 Full ablation results

Tables 5, 6 and 4 list the per-benchmark accuracies behind Figures 4, 5 and Table 2.

9.4 Worked example of the recall reward

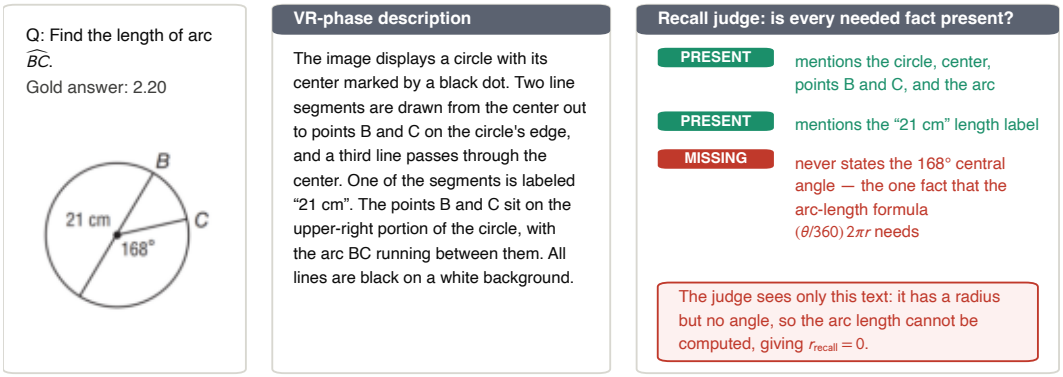


Figure 6: **The recall reward on a worked example.** *Left:* the question and image (arc length of $\widehat{BC}$). *Center:* the description produced by the VR phase. *Right:* a text-only judge, which never sees the image, checks whether the description alone contains every fact the question needs. It finds the circle, center, points and the "21 cm" label, but the description *never states the* $168°$ *central angle*—the one fact the arc-length formula requires—so the description is insufficient and $r_{\text{recall}} = 0$.

9.5 Two-pass evaluation

All results in the main paper use single-pass CoT inference. Table 7 instead evaluates each model in the two-pass format used during training: the model first describes the image and then answers from its own description, without the image. SPLIT-RL improves over the base model by $+8.1$ points on Qwen3-VL-8B and $+10.5$ on Qwen3-VL-4B, 3.9 and 8.2 points more than GRPO. The two-pass format is thus where the gains of VR and LR add up most visibly: the model must reason from its own description, so the modest VR gain measured with Kimi-K3 as the reasoner (Section 4.3.2) matters.

Table 5: **Contribution of each component** on Qwen3-VL-8B. All staged variants share the same LR stage and differ only in the rewards used in the VR stage. Δ is the change in average accuracy relative to the untrained Qwen3-VL-8B. The best result per column is shown in **bold**.

Method	MVista	MVision	MVerse	WeMath	LogicVista	DynaMath	**AVG**	Δ
Qwen3-VL-8B	77.2	63.2	63.8	79.9	60.4	74.4	69.8	—
VR → LR	79.3	65.5	68.2	84.8	63.1	77.1	73.0	+3.2
VR + Recall → LR	79.6	64.1	67.4	84.9	65.1	77.3	73.1	+3.3
VR + CLA → LR	**80.1**	61.2	**70.9**	**87.9**	65.3	76.9	73.7	+3.9
VR + CLA + Recall → LR	79.8	**69.4**	69.8	87.4	**66.0**	**79.1**	**75.3**	**+5.4**

Table 6: **Improvement of each capability** on Qwen3-VL-8B. **(a)** VR: the describer is varied and the reasoner is fixed to Kimi-K3. **(b)** LR: the describer is fixed to Kimi-K3 and the reasoner is varied. The oracle row uses Kimi-K3 for both roles and is shared by both blocks. Δ is the change in mean accuracy relative to the base model. Within each block, best per column (excluding the oracle) in **bold**, second best underlined.

Method	MVista	MVision	MVerse	WeMath	LogicVista	DynaMath	**Mean**	Δ
(a) Varying the describer (reasoner: Kimi-K3)								
Kimi-K3 (oracle)	86.4	84.9	85.9	89.8	80.1	90.6	86.3	
Qwen3-VL-8B	75.6	66.1	72.8	80.5	**66.2**	75.9	72.9	0.0
Qwen3-VL-8B (GRPO)	76.3	66.8	**74.6**	80.3	63.5	76.0	72.9	0.0
VR → LR	77.0	64.8	73.7	81.7	65.5	**78.3**	73.5	+0.6
VR + Recall → LR	76.0	64.7	74.0	81.8	64.4	75.6	72.7	−0.2
VR + CLA + Recall → LR	**79.6**	**67.4**	74.4	**83.1**	**66.2**	76.8	**74.6**	**+1.7**
(b) Varying the reasoner (describer: Kimi-K3)								
Kimi-K3 (oracle)	86.4	84.9	85.9	89.8	80.1	90.6	86.3	
Qwen3-VL-8B	80.1	63.2	69.5	83.6	65.8	87.5	74.9	0.0
Qwen3-VL-8B (GRPO)	83.0	68.1	74.0	90.6	69.1	87.2	78.7	+3.8
VR → LR	84.2	69.7	**76.6**	**91.0**	69.8	88.6	80.0	+5.1
VR + Recall → LR	82.7	69.1	74.6	89.7	**71.6**	88.8	79.4	+4.5
VR + CLA + Recall → LR	**85.5**	**72.7**	76.3	**91.0**	**71.6**	**89.3**	**81.1**	**+6.2**

Two-pass accuracy is lower than single-pass accuracy for every model, including the base model (64.0 vs. 69.8 on Qwen3-VL-8B). This drop is a property of the format rather than of training: the description is an information bottleneck, so any visual fact it omits or misstates cannot be recovered, and the reasoner cannot look at the image again. SPLIT-RL loses the least (-3.2 points on Qwen3-VL-8B, vs. -4.4 for GRPO and -5.8 for the base model; -4.1 vs. -8.6 and -9.8 on Qwen3-VL-4B), consistent with its descriptions being more self-sufficient. SPLIT-RL leads in both formats: in the two-pass format it is trained in, and in the single-pass format that GRPO is trained in. Since single-pass inference is both cheaper and more accurate, we use it for all main results.

9.6 Phase prompts

We use one fixed prompt per phase for all training and diagonistic evaluation. The VR-phase prompt asks the model to state only the visual content relevant to the question, without reasoning towards an answer; the LR-phase prompt asks the model to answer from the question and the description alone, without access to the image.

VR phase (describer) prompt

```
You are a careful visual perceiver. You will be shown an image and a
question. Your ONLY job is to produce a faithful, detailed visual
description of the image that is relevant to the question. You MUST
NOT attempt to solve, infer the answer, or perform reasoning beyond
direct observation.

# What to describe
```

```
Describe everything in the image that is relevant to the question.
Cover, when present:
- Objects, shapes, and geometric primitives (points, lines, segments,
  rays, angles, arcs, circles, polygons, curves, axes, gridlines).
- Text content exactly as it literally appears in the image
  (character-for-character, including capitalization and punctuation).
- Every numerical value and unit visible in the image, and where each
  value is placed. Aim to preserve ALL numeric tokens that appear on
  the figure, especially side lengths, angle measures, coordinates,
  tick values, and axis labels -- do not paraphrase them away.
- Variables and formulas as they are rendered on the figure. Wrap math
  expressions and single-letter variables in inline LaTeX using
  `$...$` (e.g. `$\angle ABC$`, `$x = 3$`, `$\frac{a}{b}$`,
  `$\overline{PQ}$`, `$60^{\circ}$`).
- Spatial relationships and layout using explicit positional language:
  `upper-left`, `upper-right`, `lower-left`, `lower-right`, `top`,
  `bottom`, `center`, `left of`, `right of`, `above`, `below`,
  `between`, `inside`, `outside`, `adjacent to`, `aligned with`.
  Anchor each labeled element to a position or to another element.
- Fine visual cues: `tick marks`, `hash marks`, `arrowheads`, `dashed
  line`, `dotted line`, `solid line`, `bold` / `italic` text, colors
  (only when visually distinctive: `red`, `blue`, `green`, `pink`,
  `orange`, `gray`, shading, fill patterns), congruence marks,
  right-angle squares, parallel-line ticks, decorative styling.
- Counts and ordering (e.g. "three points labeled A, B, C from left
  to right"; "four bars, tallest on the right").
- For charts/plots: axis titles, axis ranges, tick values, legend
  entries, series names, and the qualitative shape of each series
  (increasing, decreasing, peaked, flat) -- but do not compute values.
- For tables: headers, row labels, and cell entries verbatim.

# Style and calibration

- Use bulleted structure (lines beginning with `- `) for enumerable
  visual facts (labels, coordinates, measurements, series). Use short
  prose paragraphs only for overall layout summary.
- Be epistemically calibrated: use hedges such as `appears to be`,
  `looks like`, `seems`, `roughly`, `approximately` when a value,
  color, or relationship is estimated from pixels rather than read
  from an explicit label. Do NOT hedge on text/numbers that are
  printed and legible -- quote those exactly.
- Describe what is actually drawn. Do NOT invent stock phrases like
  "coordinate plane", "clockwise order", "counter-clockwise order",
  "not labeled", "no measurement", "no markings" unless that absence
  is genuinely relevant to answering the question and is a real
  feature of the image.
- If something is illegible or ambiguous, say so explicitly and, if
  possible, describe what you can see (e.g. "a two-digit number,
  first digit appears to be `4`, second is unclear").

# Output contract

- Output a SINGLE block of plain text (bullets allowed inside it).
- Target length: roughly 800-1500 characters. Shorter is fine only if
  the image is trivially simple; longer is fine only if the image is
  unusually dense. Never exceed 2500 characters. Never repeat the
  same phrase or bullet -- if you find yourself repeating, STOP.
- Do NOT output a `Final Answer`, `Answer:`, `Therefore`, a solution,
  a computed result, or any conclusion. Do NOT solve the problem.
- Do NOT introduce facts, measurements, or labels not visible in the
  image. Do NOT infer values that are not explicitly shown.

## Question:
<question>

## Visual Description:
```

LR phase (reasoner) prompt

```
You are given (a) a question, (b) a visual description of the image
that another model produced. Using ONLY the visual
description and the question, solve the problem
step by step.
```

```
Follow these rules strictly:
- Commit to ONE interpretation of the figure on your first read and
  solve under that interpretation. Do not enumerate alternative
  layouts or restart from a different reading. Do not write
  "wait", "actually", "hold on", "let me reconsider", or similar
  self-correction openers -- if you notice a mistake, fix that
  specific step and move on.
- Every quantity named in the description (a length, an angle, a
  label like "13" or "6x+7") is real and given for a reason. Use it
  as data. Do not doubt whether it is a length, an angle, or valid.
- Assume the problem is solvable. Only after you have made a
  serious attempt and can identify a specific missing fact should
  you consider refusing. Prefer a best-effort numeric answer under
  the most natural interpretation over refusing.
- Keep the reasoning short and linear: one plan, one derivation,
  one final answer.

If, after a genuine attempt, a specific fact truly is missing and
no natural interpretation lets you proceed, briefly state what is
missing and answer can't be answered.

## Question:
<question>

## Visual Description (from describer):
<description>
```

9.7 JUDGE PROMPTS

We use two judges in the VR stage, both instantiated with Claude Sonnet 4.6 (Anthropic, 2026), the same model used for answer evaluation (Appendix 9.2). The *recall* judge is text-only: it receives the question and the VR-phase description, but never the image, and decides whether the description alone contains every visual fact needed to answer the question. The CLA-GRPO judge does see the image, and labels each atomic claim with a correctness verdict, a relevance verdict and a visual type.

Recall judge prompt (VR phase)

```
You are auditing a visual description for information sufficiency AND for staying
     within its role. You are NOT given the image. You are given (a) a question that
     was asked about an image and (b) a description of that image produced by another
     model. The description's ONLY job is to state the visual facts of the image; a
     separate reasoning stage (which you do NOT see) is responsible for using those
     facts to reason about and answer the question.

Decide the score by applying BOTH rules below. The score is 0 if EITHER rule fails.

Rule A -- sufficiency:
- The description must state (explicitly or unambiguously) every visual fact a
     competent reasoner would need to answer the question. If at least one required
     piece of visual information is missing, ambiguous, or only partially specified,
     the description is insufficient.

Rule B -- role (no reasoning / no answer):
- The description must ONLY describe visual elements. It must NOT do the reasoning the
     later stage is responsible for, and it must NOT state or imply the answer to the
     question. This includes: giving the final answer, computing or deriving a result,
     drawing a conclusion, comparing/selecting among options to decide which is correct
     , or narrating step-by-step reasoning toward the answer. Reporting a visual fact
     that is directly readable from the image (e.g. a printed label, a number written
     in the figure, a color, a count of visible objects) is fine and expected;
     performing the calculation or inference that answers the question is NOT.

Scoring:
- score = 1 -- the description satisfies Rule A (sufficient) AND Rule B (purely
     descriptive, no reasoning and no answer).
- score = 0 -- the description fails Rule A (insufficient) OR fails Rule B (it contains
      the answer or performs the reasoning). If the description contains or derives the
      answer, score 0 even if it is otherwise sufficient.

Important:
- Do NOT judge whether the description is correct -- only sufficiency (Rule A) and role
      (Rule B). Even a wrong description can be sufficient.
```

```
- Do NOT judge whether the question can be answered without an image (text-only) --
    assume the question genuinely depends on visual content.
- Numerical values that are literally printed/visible in the image are visual facts and
     are allowed; a calculation, comparison, or inference performed on those values to
     reach the answer is reasoning and is NOT allowed.

Question:
<question>

Description:
<description>

Respond with ONLY a single JSON object of this shape:

{"score": 0, "rationale": "<=1 short sentence"}

No markdown, no commentary outside the JSON object, no code fences.
```

CLA-GRPO judge prompt (claim verification)

```
You are verifying the factual accuracy of a visual description against the image it
    describes. You ARE given the image. You are also given (a) a question that was
    asked about the image and (b) a list of components -- individual claims extracted
    from a model's description of the image. Judge EACH component independently.

For each component, decide:
- correct: 1 if the component is a faithful, accurate statement about the image (it
    does not contradict what is visible and is supported by the image); 0 if it is
    false, hallucinated, or contradicts the image.
- relevance: HIGH if the component states visual information that matters for answering
     the question; LOW if it is accurate/inaccurate but tangential to the question.
- component_type: the single category (from the list below) that best describes what
    kind of visual claim the component makes.
- rationale: a terse justification of at most 12 words (keep it short).

Allowed component_type values (choose exactly one, verbatim): object_presence, count,
    color_material, spatial_position, text_label, geometric_relation, line_style,
    absence, scene_context, other

Category meanings (guidance for the default categories; if a name above is not listed
    here, judge by its plain meaning):
- object_presence: an object / shape / curve exists, or its identity (e.g. "a circle",
    "a red cylinder", "a lizard").
- count: a number or quantity of items (e.g. "three points", "ten circles", "there are
    two segments").
- color_material: color, shade, material, or finish (e.g. red, gray, matte, metallic,
    reflective).
- spatial_position: location, orientation, or relative position (e.g. left, above,
    center, quadrant, aligned).
- text_label: labels, tick values, symbols, captions, or axis names (e.g. "labeled A",
    tick "-3d", overline/angle notation on segments).
- geometric_relation: a quantitative or geometric relation (e.g. parallel,
    perpendicular, angle, equal, congruent, bisects, coordinate, slope, connects,
    intersects, forms a triangle).
- line_style: stroke or fill attributes (e.g. solid, dashed, thick, filled, arrowhead,
    hatching).
- absence: a negated or absent feature (e.g. "no numerical values", "no gridlines", "no
     colors").
- scene_context: background, surface, lighting, shadow, or gridlines.
- other: anything that fits none of the above.

Rules:
- Judge correctness ONLY against the image, not against the question's answer. A
    component can be correct (faithful to the image) even if it is irrelevant to the
    question.
- Judge each component on its own; do not let one component's error affect another's
    score.
- If a component is vague or unverifiable from the image, mark correct = 0.
- Pick the single best-fitting component_type. If a component makes several claims,
    pick the type of its PRIMARY claim. If none fit well, use "other".

Question:
<question>

Components (index in brackets):
```

Table 7: **Two-phase (self-loop) evaluation.** The model first describes the image and then answers from its own description, without the image, as in training. Δ is the change in mean accuracy relative to the base model of the same size. Best per column within each backbone in **bold**.

Method	MVista	MVision	MVerse	WeMath	LogicVista	DynaMath	**Mean**	Δ
Qwen3-VL-8B								
Base	69.1	54.3	58.1	73.9	57.7	71.1	64.0	–
GRPO	73.4	57.2	63.7	80.3	62.4	72.2	68.2	+4.2
SPLIT-RL	**76.6**	**62.8**	**66.4**	**85.9**	**65.3**	**75.9**	**72.1**	**+8.1**
Qwen3-VL-4B								
Base	68.4	45.7	51.9	64.1	50.8	70.6	58.6	–
GRPO	69.1	49.0	58.5	68.6	50.3	69.7	60.9	+2.3
SPLIT-RL	**74.3**	**55.6**	**64.1**	**84.3**	**61.1**	**75.0**	**69.1**	**+10.5**

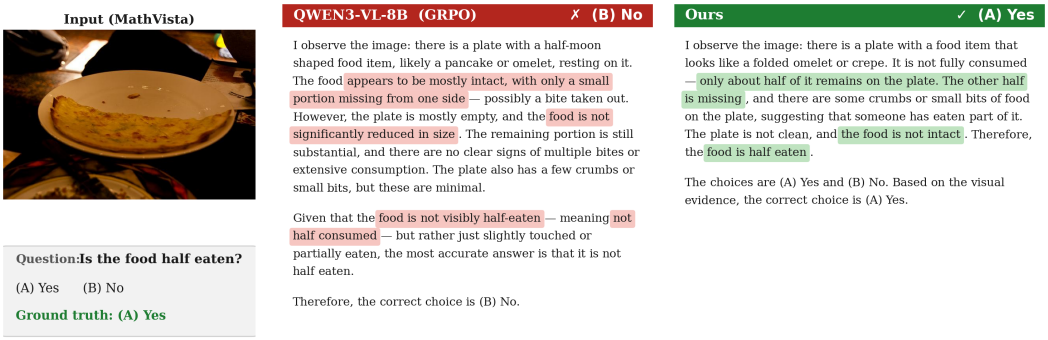


Figure 7: **Qualitative example (MathVista).** *Is the food half eaten?* The GRPO baseline hedges about the food being "mostly intact" and answers **(B) No** (✗). Our model identifies that only half the food remains and answers **(A) Yes** (✓), matching the ground truth.

```
<indexed list of atomic claims>

Respond with ONLY a single JSON object of this exact shape:

{"components": [{"index": 0, "correct": 0, "relevance": "LOW", "component_type": "other
    ", "rationale": "<=1 short sentence"}]}

Include exactly one entry per input component, using the same index. No markdown, no
    commentary outside the JSON object, no code fences.
```

9.8 Additional Qualitative Examples

we show addition quanlitative example in figure 7.

9.9 Limitations and Future Work

SPLIT-RL relies on LLM judges for the recall reward and CLA-GRPO, which adds training cost, and its descriptions still trail those of a frontier model, so perception remains the main source of headroom. With the vision tower frozen, SPLIT-RL's gains over GRPO shrink as backbones improve, leaving its effectiveness at frontier scale and the need for harder perception-bound data open. Natural next steps are to extend the approach to other domains such as video, and to settings with multiple interleaved rounds of perception and reasoning.