RIGHT ANSWERS FOR THE RIGHT REASONS? BRIDGING REASONING UTILITY AND VISUAL DEPENDENCY IN MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has advanced multimodal reasoning, but final-answer supervision cannot distinguish sound reasoning from flawed trajectories that happen to produce correct answers. Our model-based and human assessments reveal substantial perception and reasoning errors among answer-correct responses, motivating feedback that differentiates successful trajectories. We propose **C**entered-**A**daptive **R**esidual **V**isual **T**ransport (CA-RVT), a design that evaluates reasoning traces using the policy's own answer-confidence probes. CA-RVT measures how each trace changes reference-answer confidence with and without direct image access, relative to trace-free baselines. It combines multimodal reasoning utility with a conservative cross-view residual and adapts the latter to visual dependency across training problems. By reusing the policy for confidence scoring, it requires neither process annotations nor external reward or judge models during training, keeping the post-training pipeline lightweight. With Qwen2.5-VL-7B-Instruct post-trained on only 2.1K Geometry3K problems, CA-RVT achieves competitive performance across mathematical reasoning and general multimodal benchmarks. It attains the highest four-benchmark mathematical average among the compared open-source models with complete results, and the best results on all seven general multimodal benchmarks in our comparison. On questions answered correctly by all audited models, it also achieves lower judged spurious-correctness rates. Additional experiments across training datasets and different backbones further support its applicability. Code will be released here.

1 INTRODUCTION

Multimodal large language models (MLLMs) must combine accurate perception with reliable inference to reason about visual inputs. Mathematical problems involving diagrams and charts challenge this ability (Lu et al., 2024; Zhang et al., 2024; Wang et al., 2024a): even a valid calculation can fail when it uses misread visual evidence. Reinforcement learning with verifiable rewards (RLVR) offers a scalable way to improve reasoning through final-answer supervision (Guo et al., 2025; Meng et al., 2025; Huang et al., 2026b). However, an incorrect derivation, a misread diagram, or a lucky guess can receive the same outcome reward as a sound solution. Improving answer accuracy therefore raises a question: can multimodal RL also favor the reasoning that supports those answers?

Figure 1(A) makes this gap concrete. One rollout makes algebraic errors and selects the correct option despite its contradictory calculation; another misreads a geometric diagram and applies an invalid relation, yet also returns the correct choice. To examine this phenomenon, we sample eight rollouts per Geometry3K question (Lu et al., 2021) from Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and retain 9,285 answer-correct trajectories. A Qwen2.5-VL-72B-Instruct judge labels only **62.0%** as ideal, with 34.0% containing combined perception–reasoning errors and 3.2% reasoning-only errors. In a separate human assessment of 500 answer-correct trajectories, **59.2%** are judged ideal (Figure 1(B)). Both assessments reveal a training opportunity hidden within successful responses: outcome rewards recognize that an answer is correct, but do not distinguish the quality of the reasoning that produced it. In group-relative optimization, this leaves useful differences among answer-correct trajectories unavailable to the outcome signal.

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.

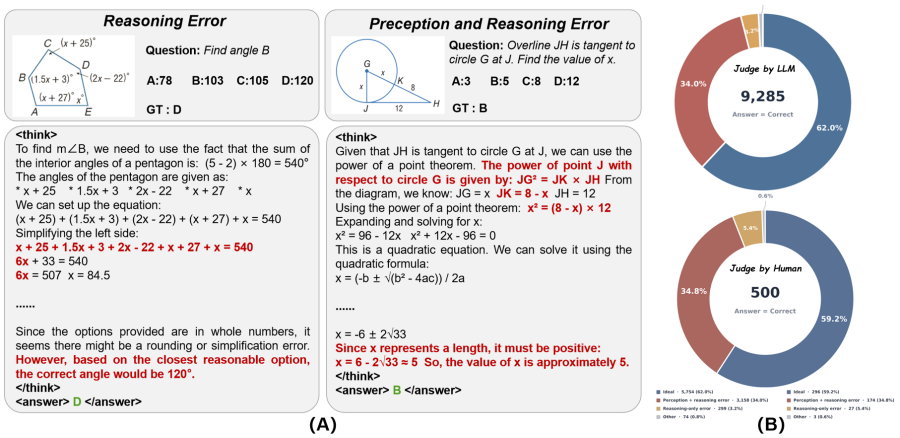


Figure 1: **Right answers can conceal wrong reasons. (A)** Answer-correct rollouts with reasoning-only errors (left) and combined perception–reasoning errors (right); red marks errors and green marks correct final choices. **(B)** Only 62.0% of 9,285 model-assessed and 59.2% of 500 human-assessed answer-correct trajectories are judged ideal.

Even within an all-correct group, traces may differ in their use of visual evidence. We therefore seek a reward that preserves outcome supervision while distinguishing the support supplied by each trace.

Process reward models and learned thinking rewards introduce finer-grained supervision (Wang et al., 2025a; Fan et al., 2026); probability-based rewards also use the policy itself to evaluate answer support (Zhou et al., 2025; Yu et al., 2025). For multimodal reasoning, a further distinction concerns the relationship between that support and the image. We ask two connected questions: *does conditioning on a trace improve support for the reference answer, and does that support persist when the probe no longer has direct image access?* The second question evaluates information carried by the same multimodally generated trace, including visual facts expressed in text.

Image-present probing evaluates the trace with the original evidence, whereas image-absent probing tests support accessible through the trace itself. The cross-view residual requires positive gains in both views, preventing one from compensating for absent support in the other; ablations test whether this constraint improves learning.

We propose **Centered-Adaptive Residual Visual Transport (CA-RVT)**. Four no-gradient confidence probes evaluate each trace with and without the image against a trace-free baseline in each view. Signed multimodal utility distinguishes increases from decreases in reference-answer confidence. A conservative cross-view residual grants additional credit only for positive utility shared by both views. A visual-dependency gate relates this residual to the image's contribution to trace-free confidence, and centered-adaptive calibration adjusts its strength relative to other problems in the rollout batch. All rollouts retain the image; removal is restricted to probing. This design refines rewards among answer-correct trajectories without process annotations or a separately trained reward model.

With post-training on only 2.1K Geometry3K problems, CA-RVT reaches **56.13%** average accuracy across four mathematical benchmarks, improving over the base model and GRPO by **5.53** and **2.73** percentage points, respectively. It also improves all seven evaluated general multimodal benchmarks. Training incurs a 15.1% increase in GPU time over matched GRPO, while inference requires only the trained policy, with no auxiliary scorer or probe evaluations (Appendix E.2). The reasoning-quality audit and component ablations examine whether these gains accompany better-supported successful responses and depend on the proposed reward design (Section 5). Our contributions are threefold:

- We identify perception and reasoning errors hidden within answer-correct multimodal trajectories through model-based and human assessments.
- We introduce CA-RVT, combining signed multimodal utility, conservative cross-view support, and centered-adaptive visual calibration. Its lightweight post-training uses a small training set, without external reward models or process annotations.
- We demonstrate gains in answer accuracy and judged reasoning quality, with additional comparisons across training datasets and different backbones. A fixed-pair probe audit further examines supervision beyond answer correctness.

2 Related Work

2.1 Reinforcement Learning for Multimodal Reasoning

RLVR for language reasoning (Guo et al., 2025) has motivated multimodal post-training with verifiable rewards. Visual-RFT targets perception and grounding (Liu et al., 2025), while MM-Eureka elicits multimodal mathematical reasoning without a supervised cold start (Meng et al., 2025). R1-OneVision and Vision-R1 combine reasoning-data construction, supervised initialization, and RL (Yang et al., 2025; Huang et al., 2026b). Complementary work studies iterative SFT–RL cycles (Deng et al., 2025), generalizable R1-style training (Shen et al., 2025), staged RL (Chen et al., 2025b), and direct RL with perception and cognition signals (Chen et al., 2025a). These approaches establish effective post-training pipelines. Our focus is the trajectory-level reward used within group-relative optimization.

2.2 Beyond Outcome Rewards

Process-aware methods evaluate intermediate reasoning through step-wise rewards, as in R1-VL (Zhang et al., 2025), or a learned multimodal process reward model, as in VisualPRM (Wang et al., 2025a). SophiaVL-R1 uses holistic thinking rewards with trustworthiness weighting and annealing (Fan et al., 2026). Perception-R1 extracts visual annotations from reference trajectories and rewards their consistency with generated responses (Xiao et al., 2026).

Policy-internal rewards use reference-answer likelihood and baseline-subtracted probability gains to provide feedback on generated reasoning (Zhou et al., 2025; Yu et al., 2025; Kwiatkowski et al., 2026). Vision-SR1 explicitly separates visual perception from subsequent reasoning and derives a self-reward from whether the policy can answer correctly using its generated perception without the image (Li et al., 2025). CA-RVT measures reference-answer confidence gains for the same reasoning trace in image-present and image-absent contexts, each relative to its trace-free baseline. It combines signed multimodal utility with a conservative cross-view residual and applies centered-adaptive calibration based on estimated visual dependency.

3 Preliminaries

Multimodal reasoning. Let $\mathcal{D} = \{(Q_j, I_j, y_j^\star)\}_{j=1}^N$ contain a question, an image, and a reference answer for each problem. Given $x_j = (Q_j, I_j)$, the policy generates G responses $r_{j,i} = (t_{j,i}, o_{j,i})$, where $t_{j,i}$ is the trace inside `<think>` and $o_{j,i}$ is the final answer inside `<answer>`. The outcome reward is

$$R_{j,i}^{\text{ans}} = \mathbb{I}[o_{j,i} \equiv y_j^\star], \tag{1}$$

where $\equiv$ denotes task-specific answer equivalence. A format reward $R_{j,i}^{\text{fmt}}$ encourages the required tag structure. Throughout the paper, j indexes problems and i indexes responses within a problem; we omit j when discussing a single problem.

Group Relative Policy Optimization. GRPO (Shao et al., 2024) samples each response group from $\pi_{\theta_{\text{old}}}$ and standardizes its scalar rewards:

$$\widehat{A}_{j,i} = \frac{R_{j,i} - \text{mean}_{k=1}^G R_{j,k}}{\text{std}_{k=1}^G R_{j,k} + \epsilon}. \tag{2}$$

The policy maximizes a clipped surrogate over response tokens,

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|r_{j,i}|}\sum_{\ell=1}^{|r_{j,i}|}\mathcal{L}_{\text{clip}}(\rho_{j,i,\ell}, \widehat{A}_{j,i})\right], \tag{3}$$

where $\rho_{j,i,\ell} = \pi_\theta(r_{j,i,\ell} \mid x_j, r_{j,i,<\ell}) / \pi_{\theta_{\text{old}}}(r_{j,i,\ell} \mid x_j, r_{j,i,<\ell})$. The inherited implementation uses asymmetric ratio clipping and dual clipping for negative advantages; Appendix E.3 specifies $\mathcal{L}_{\text{clip}}$ and its settings. We omit the KL penalty. CA-RVT changes the rollout reward, not this group-relative policy-optimization procedure.

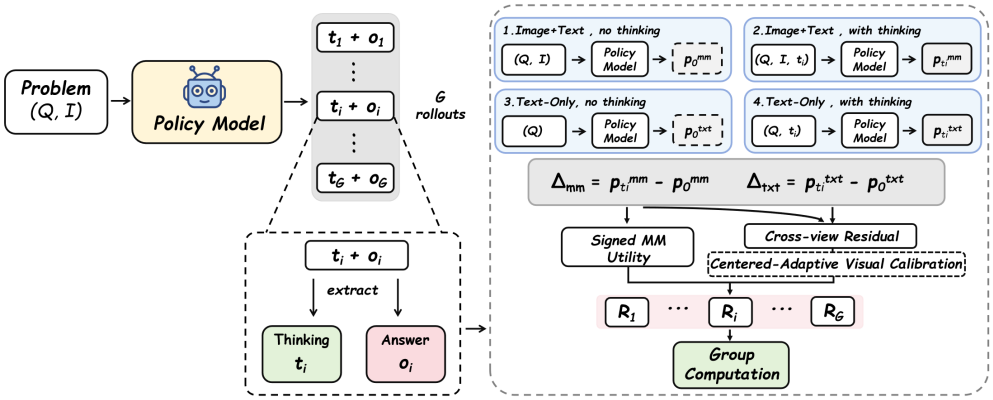


Figure 2: **Overview of CA-RVT.** The policy samples G responses per problem, each containing a reasoning trace t_i and an answer o_i. Four no-gradient probes reuse the policy to measure reference-answer confidence with and without the trace in image-present and image-absent contexts. Signed multimodal utility and a visually gated cross-view residual provide auxiliary rewards. Centered-adaptive visual calibration rescales only the residual according to each problem's visual dependency relative to the mean over the full rollout batch. The auxiliary rewards are gated by answer correctness and combined with outcome and format rewards before computing GRPO advantages. Image removal is restricted to probing.

4 METHOD

4.1 OVERVIEW

CA-RVT rewards traces that support the correct answer in the original multimodal context and gives additional credit when positive utility persists without direct access to the image. Figure 2 illustrates rollout generation, cross-view probing, and reward construction. A problem-level visual-dependency gate controls the cross-view residual, which is further rescaled by centered-adaptive visual calibration relative to the mean visual dependency across the full rollout batch. An answer-correctness gate restricts auxiliary trace rewards to successful responses. All auxiliary scores are treated as constants during the policy update. The design separates three roles: signed multimodal utility evaluates whether reasoning helps the original task; cross-view support tests whether useful information is retained through the trace; and visual calibration adjusts the residual according to the problem's estimated image dependence.

4.2 CROSS-VIEW ANSWER-CONFIDENCE PROBING

For one problem $(Q, I, y^\star)$, write $p_\theta(y^\star \mid c)$ for the policy's answer-confidence score under probe context c. The sampled trace t_i is held fixed across the two views:

$$p_0^{\mathrm{mm}} = p_\theta(y^\star \mid Q, I), \qquad p_{t_i}^{\mathrm{mm}} = p_\theta(y^\star \mid Q, I, t_i), \tag{4}$$

$$p_0^{\mathrm{txt}} = p_\theta(y^\star \mid Q), \qquad p_{t_i}^{\mathrm{txt}} = p_\theta(y^\star \mid Q, t_i). \tag{5}$$

Here θ denotes the policy state used to construct rewards for the sampled batch; probe scores are detached from optimization. The trace always originates from (Q, I): the text-only view removes the image from the *probe*, not from the generation that produced the trace. The two trace-free scores are shared by all responses to the same problem.

What the confidence probe computes. The probe consumes the complete extracted `<think>` trace: the configured prefix ratio is 1.0, with no multi-prefix averaging. For an answer continuation $a = (a_1, \ldots, a_L)$, we compute its mean teacher-forced token log-likelihood,

$$\ell_\theta(a \mid c) = \frac{1}{L} \sum_{k=1}^{L} \log \pi_\theta(a_k \mid c, a_{<k}). \tag{6}$$

For the multiple-choice training problems used here, let $\mathcal{A}$ contain the candidate option letters and $\mathcal{A}^\star \subseteq \mathcal{A}$ the accepted options. The normalized answer-confidence score is

$$p_\theta(y^\star \mid c) = \sum_{a \in \mathcal{A}^\star} \frac{\exp \ell_\theta(a \mid c)}{\sum_{a' \in \mathcal{A}} \exp \ell_\theta(a' \mid c)}. \tag{7}$$

Equation (7) instantiates the confidence score for our multiple-choice training data; the cross-view reward operates on confidence differences across the four probe contexts.

All four conditions share the same textual template and scoring rule; only the image and trace fields differ. The reference answer is a scoring target, not supplied as a known solution in the prompt. Scoring settings and exact templates appear in Appendices E and F, respectively.

4.3 Signed Multimodal Reasoning Utility

The first signal measures the trace's effect on correct-answer confidence while retaining the image:

$$\Delta_{\mathrm{mm},i} = p_{t_i}^{\mathrm{mm}} - p_0^{\mathrm{mm}}, \qquad R_i^{\mathrm{MM}} = \mathrm{clip}(\Delta_{\mathrm{mm},i}, -c_{\mathrm{mm}}, c_{\mathrm{mm}}). \tag{8}$$

A positive value indicates greater confidence after conditioning on the trace; a negative value indicates reduced confidence even when the sampled final answer is correct. Clipping bounds the influence of extreme changes without discarding the sign. We refer to this component as signed multimodal utility (MM). The *MM utility only* ablation retains its answer-gated contribution together with the outcome and format rewards. This component evaluates functional support for the reference answer and distinguishes confidence-increasing from confidence-reducing traces among successful responses.

4.4 Conservative Cross-View Residual

To test whether answer-supporting information remains accessible through the trace without direct access to the image, define

$$\Delta_{\mathrm{txt},i} = p_{t_i}^{\mathrm{txt}} - p_0^{\mathrm{txt}}, \qquad T_i = \min([\Delta_{\mathrm{mm},i}]_+, [\Delta_{\mathrm{txt},i}]_+)\,, \tag{9}$$

where $[u]_+ = \max(u, 0)$. The minimum requires positive utility in both views: a large gain in one cannot compensate for absent support in the other. The image-absent probe measures the answer support carried by the trace, including visual information and deductions expressed in text. To relate this support to the image, we condition the residual on the image's estimated contribution to the trace-free score:

$$g_{\mathrm{vis}} = \mathrm{clip}\left(\frac{p_0^{\mathrm{mm}} - p_0^{\mathrm{txt}}}{\tau_{\mathrm{vis}}}, 0, 1\right), \qquad R_i^{\mathrm{CV}} = \min(c_{\mathrm{cv}}, g_{\mathrm{vis}} T_i). \tag{10}$$

Here $\tau_{\mathrm{vis}} > 0$ sets the gate scale, and $c_{\mathrm{cv}} > 0$ caps the residual. A non-positive baseline image gain closes the gate; a positive gain opens it progressively, reaching one at τ_{vis}. The gate therefore estimates *model-relative visual dependency*.

The asymmetry between the two components is deliberate. R_i^{MM} can reduce a successful rollout's reward, whereas $R_i^{\mathrm{CV}} \geq 0$ adds credit only under stricter cross-view conditions. Here, *transport* denotes answer-supporting information retained through the text trace after image removal. The minimum and the visual gate address different ambiguities: the former requires joint utility, while the latter suppresses the bonus on problems for which the image adds little trace-free confidence. Together, they define a conservative proxy for image-related answer support that complements signed multimodal utility.

4.5 Centered-Adaptive Visual Calibration

The visual gate measures absolute baseline image support. Centered-adaptive calibration (CA) adjusts the CV residual relative to other problems in the current rollout batch. We treat centering and adaptation as one transformation: centering defines the neutral reference, while adaptation sets each problem's deviation from it. For B problems, let

$$\bar{g}_{\mathrm{vis}} = \frac{1}{B} \sum_{j=1}^{B} g_{\mathrm{vis}}^{(j)}, \qquad s_j = \mathrm{clip}\Big(1 + \kappa\big(g_{\mathrm{vis}}^{(j)} - \bar{g}_{\mathrm{vis}}\big), s_{\min}, s_{\max}\Big)\,, \tag{11}$$

with $\kappa \geq 0$ and $0 < s_{\min} \leq 1 \leq s_{\max}$. The implementation takes the mean of finite gates over the *entire gathered rollout batch*, before answer-correctness gating, rather than separately within each question or GPU worker. With equal-size response groups and finite gates, this equals the problem-level mean in Equation (11). The same s_j is used for all responses to problem j; centering within one question would instead give $s_j = 1$ and remove the adaptation. Above-average visual dependency increases the multiplier, while below-average dependency decreases it, subject to clipping. Unlike an additional factor restricted to $[0, 1]$, this construction allows both amplification and attenuation around one.

Centering gives the *unclipped multipliers* a batch mean of one. Since s_j scales only the CV residual, it adjusts that component's contribution to the within-group reward mixture. Appendix C derives the effects of clipping, gate–residual covariance, and GRPO normalization.

4.6 Correctness-Gated Reward

For rollout i of problem j, CA-RVT uses

$$R_{j,i} = R^{\text{ans}}_{j,i} + \lambda_{\text{fmt}} R^{\text{fmt}}_{j,i} + R^{\text{ans}}_{j,i} \left(\lambda_{\text{mm}} R^{\text{MM}}_{j,i} + \lambda_{\text{cv}} s_j R^{\text{CV}}_{j,i} \right). \tag{12}$$

The answer gate ensures that an incorrect response receives no auxiliary trace-utility reward even when its trace increases reference-answer confidence. Among successful responses, signed multimodal utility measures support in the original context, and the residual supplies additional credit when that support is positive in both views and the baseline image gain is positive. We compute advantages from the rewards using Equation (2) and update the policy using Equation (3). With the configured coefficients (Appendix E.1), correct-response rewards remain above incorrect-response rewards even after the signed utility penalty. Within an all-correct group, however, differences in trace utility can still produce nonzero advantages when outcome and format rewards are tied. This is the distinction unavailable to outcome-only supervision. All probe computations are training-time operations; inference uses the resulting policy without these auxiliary evaluations.

5 Experiments

5.1 Experimental Setup

Training and evaluation. Our main model starts from Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and is post-trained on 2.1K Geometry3K problems (Lu et al., 2021). We evaluate mathematical reasoning on MathVerse testmini, MathVision test, MathVista testmini, and WeMath testmini (Zhang et al., 2024; Wang et al., 2024a; Lu et al., 2024; Qiao et al., 2025). MathVerse includes its five image-conditioned variants, excluding Text Only; WeMath is evaluated in the multiple-choice setting. The reported mathematical average is the unweighted mean of the four benchmark scores. We also evaluate seven general multimodal benchmarks to examine transfer beyond the training domain. We use GRPO with a global problem batch of 32 and eight responses per question, without KL regularization. Benchmark descriptions and baseline references are in Appendix A; implementation settings and prompt templates are provided in Appendices E and F, respectively.

Comparisons. We compare with general-purpose and reasoning-oriented open-source models, and include two proprietary models as reference points. The external models cover different training data and post-training pipelines; the GRPO and component ablations isolate reward-design choices within our training setting. For supported general benchmarks, we use the generation-based VLMEvalKit evaluation pipeline (Duan et al., 2024).

5.2 Mathematical Reasoning and Transfer

Improved mathematical reasoning with a small training set. Table 1 reports **56.13%** average accuracy for CA-RVT, the highest among the compared open-source models with scores on all four benchmarks. It exceeds the base model and GRPO by **5.53** and **2.73** points, respectively, and improves over GRPO on every benchmark. The GRPO comparison indicates that correct-answer supervision leaves room for more informative trajectory feedback on the same 2.1K training problems. CA-RVT leads the open-source comparison on MathVerse, MathVision, and WeMath; MM-Eureka remains strongest on MathVista.

Table 1: Mathematical reasoning results (accuracy, %). All open-source models are re-evaluated using the same evaluation protocol. **Bold** denotes the best result among the compared open-source models. Avg. is the unweighted mean of the displayed benchmark scores, rounded to two decimals. Data lists post-training set sizes; Results for R1-VL-7B on MathVision and WeMath are omitted ("–") because both benchmarks were included in its training data.

Model	Data	Math Benchmarks				Avg.
		MathVerse testmini	**MathVision** test	**MathVista** testmini	**WeMath** testmini	
Proprietary MLLMs						
GPT-4o	–	50.2	30.4	63.8	68.8	53.30
Claude-3.7-Sonnet	–	52.0	41.3	66.8	72.6	58.18
Open-Source MLLMs						
Qwen2-VL-7B-Instruct	–	31.3	14.8	58.0	42.5	36.65
Qwen2.5-VL-7B-Instruct	–	47.8	25.1	68.1	61.4	50.60
Qwen2.5-VL-7B + GRPO	2.1K	50.8	27.3	68.2	67.3	53.40
R1-VL-7B	260K	41.1	–	62.7	–	–
R1-OneVision-7B	155K	46.3	21.9	65.2	62.0	48.85
OpenVLThinker-7B	25K	47.6	24.2	71.2	66.1	52.28
DRIFT-VL-7B	4K	52.6	26.8	69.4	68.6	54.35
VLAA-Thinker-7B	25K	51.5	26.7	71.1	66.2	53.88
Vision-SR1-7B	47K	47.6	28.7	70.2	65.8	53.08
SophiaVL-R1-7B	130K	49.6	26.6	70.8	66.9	53.48
MM-Eureka-7B	15K	51.9	28.7	**72.5**	64.9	54.50
CA-RVT (ours)	**2.1K**	**53.6**	**29.0**	71.2	**70.7**	**56.13**

Table 2: General multimodal results (accuracy, %). HR-4K: HR-Bench4K; V*: V*Bench; CountQA: CountBenchQA; RWQA: RealWorldQA. MMMU uses dev/val and the other benchmarks use their reported test settings. **Bold** marks the highest reported score in each column.

Model	Perception-Centric			Visual Understanding			General Reasoning
	HR-4K test	**V*** test	**CountQA** test	**AI2D** test	**RWQA** test	**ChartQA** test	**MMMU** dev & val
Qwen2.5-VL-7B-Instruct	68.5	76.9	89.3	85.0	69.0	87.2	51.3
Qwen2.5-VL-7B + GRPO	69.0	75.4	89.3	85.3	69.2	87.3	52.4
VLAA-Thinker-7B	52.1	55.5	86.4	84.4	66.0	86.0	51.3
SophiaVL-R1-7B	69.4	75.9	87.7	85.1	68.4	87.4	51.0
MM-Eureka-7B	50.9	56.5	86.4	84.8	66.1	85.6	54.1
CA-RVT (ours)	**70.0**	**77.5**	**89.7**	**85.5**	**69.7**	**87.5**	**54.7**

Reasoning gains with preserved visual capabilities. Several reasoning-oriented models improve mathematical accuracy yet fall below the base model on perception and visual-understanding benchmarks (Table 2). This may reflect specialization toward reasoning tasks, where answer-oriented optimization provides insufficient feedback on the use of visual evidence. CA-RVT couples multimodal trace utility with a cross-view residual calibrated by visual dependency, encouraging traces that support the correct answer while retaining task-relevant visual information. Its improvements over both the base model and GRPO across all seven benchmarks are consistent with better preservation of visual capabilities during reasoning post-training.

5.3 Contributions of Utility and Calibration

Table 3 organizes CA-RVT into signed multimodal utility (MM), the visually gated cross-view residual (CV), and centered-adaptive calibration (CA). The single-term variants retain one answer-

Table 3: Reward-component ablations. MM: signed multimodal utility; CV: visually gated cross-view residual; CA: centered-adaptive calibration. Scores are accuracy (%); best results per column are bold. The GRPO baseline is reported in Table 1.

Configuration	MM	CV	CA	MathVerse	MathVision	MathVista	WeMath	Avg.
MM utility only	✓	–	–	52.4	28.0	71.3	70.0	55.43
CV residual only	–	✓	–	50.8	28.4	70.3	68.8	54.58
MM+CV	✓	✓	–	52.2	28.6	**71.6**	69.8	55.55
CA-RVT	✓	✓	✓	**53.6**	**29.0**	71.2	**70.7**	**56.13**

Table 4: Residual aggregation ablations, all retaining signed MM utility. The image-present residual uses the positive multimodal gain; harmonic mean and maximum replace the positive-part minimum in the CV residual. Scores are accuracy (%); Avg. is the unweighted mean across the four benchmarks. Bold denotes the best result in each column.

Residual variant	MathVerse	MathVision	MathVista	WeMath	Avg.
Image-present residual	51.8	28.9	71.0	69.4	55.28
Harmonic-mean residual	52.1	28.4	69.9	69.3	54.93
Maximum residual	51.8	27.7	**71.6**	69.1	55.05
CA-RVT (minimum)	**53.6**	**29.0**	71.2	**70.7**	**56.13**

gated auxiliary reward; MM+CV uses both with fixed scale $s_j = 1$. All variants retain outcome and format rewards. An additional uncentered-calibration comparison is given in Appendix C.1.

Cross-view support complements original-context utility. MM utility alone reaches 55.43, compared with 54.58 for CV alone. The stronger MM average favors retaining a direct test of whether a trace helps the original multimodal task alongside cross-view support. CV alone is stronger on MathVision, indicating different performance profiles. The fixed combination reaches 55.55%, compared with 55.43% for MM alone.

Calibration controls the balance between auxiliary signals. The full method improves the fixed combination by 0.58 average point, chiefly through MathVerse and WeMath, with a 0.4-point MathVista trade-off. Both configurations use the same joint-support constraint; the difference is the problem-dependent strength of its residual. This supports further calibrating the residual beyond its absolute visual gate. Centering defines the multiplier's neutral reference, and adaptation redistributes its strength around that reference; they are evaluated here as one calibration module.

Why limit the residual to jointly supported utility? Table 4 changes the residual while retaining signed MM utility. An image-present residual omits the TXT requirement; a maximum allows either view to determine the bonus; and a harmonic mean can exceed the weaker gain when both are positive. The minimum instead prevents stronger utility in one view from compensating for weaker support in the other. It achieves the highest average, **56.13**, compared with 55.28 for the image-present residual, 54.93 for the harmonic mean, and 55.05 for the maximum. The maximum is stronger on MathVista but weaker on the other three tasks. These comparisons support using the weaker positive gain to limit the cross-view residual under the tested training configuration. Halving or doubling λ_{cv} or τ_{vis} one at a time yields average accuracies of 55.1–55.7%, compared with 56.13% for the default setting (Appendix D.6).

5.4 Reasoning Quality Beyond Answer Accuracy

Comparing reasoning on the same successful problems. We compare CA-RVT, Qwen2.5-VL-7B-Instruct, SophiaVL-R1-7B, and MM-Eureka-7B using Qwen2.5-VL-72B-Instruct as the trajectory judge. The judge is used only for evaluation and supplies no training rewards. The shared-question protocol holds question identity and final-answer correctness fixed; spurious correctness counts judged perception or reasoning errors, excluding overthinking alone (Appendix B). CA-RVT has the lowest rates on all four shared subsets (Figure 3, left). Since all compared answers are correct on these questions, this result reveals differences in assessed reasoning quality beyond outcome accuracy.

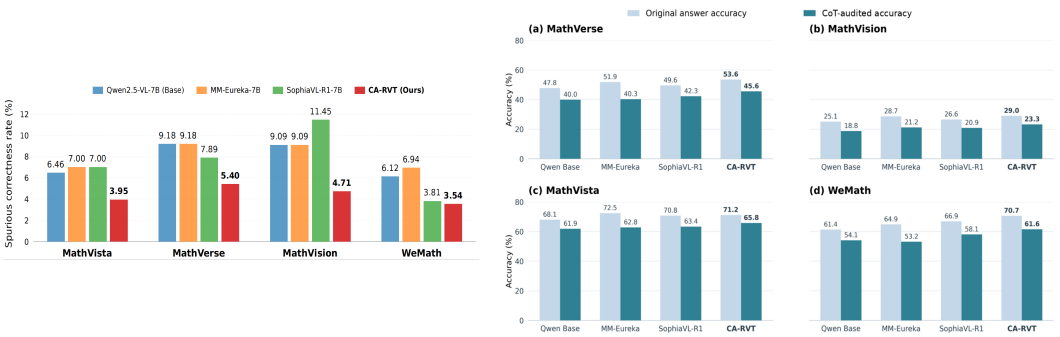


Figure 3: **Reasoning quality under two evaluation protocols. Left:** Spurious-correctness rates on questions answered correctly by all four models (lower is better). The shared sets contain 557 MathVista, 1,166 MathVerse, 297 MathVision, and 735 WeMath questions. **Right:** Answer accuracy and CoT-audited accuracy computed from the same saved responses for each model and benchmark (higher is better). Both metrics use the full evaluation set as the denominator. CoT auditing excludes answer-correct responses with judge-identified decisive perception or reasoning errors in the final endorsed solution; uncertain cases are retained.

Auditing each model's own successful responses. For each model and benchmark, we compute answer accuracy and CoT-audited accuracy from the same saved responses. The latter additionally excludes answer-correct responses with judge-identified decisive perception or reasoning errors in the final endorsed solution. Both metrics use the full evaluation set as the denominator, so their difference measures the proportion of questions answered correctly despite such errors. Explicitly corrected mistakes and overthinking alone do not incur a deduction. CA-RVT achieves the highest CoT-audited accuracy on all four benchmarks (Figure 3, right): 45.6, 23.3, 65.8, and 61.6 on MathVerse, MathVision, MathVista, and WeMath, respectively.

Does the utility score distinguish more than answer declarations? We examine 100 fixed same-question pairs of answer-correct traces with different judged process quality (Appendix D.5). Using the original base checkpoint as a frozen probe avoids fitting the probe to CA-RVT's outputs; matching questions and correctness isolates distinctions unavailable to binary outcome rewards. The auxiliary score achieves 76.5% pairwise ranking accuracy against the 50% outcome-only tie baseline. Removing detected explicit answer declarations preserves discrimination (79.5%), whereas retaining only those declarations yields 50.0%. The same pattern holds among pairs modified by the intervention, showing that the utility score retains discrimination between traces with different judge-assessed process quality after detected explicit answer declarations are removed. Additional qualitative comparisons across models appear in Appendix G.

6 Conclusion and Future Work

In this paper, we introduced CA-RVT to address the mismatch between answer correctness and reasoning quality in multimodal reinforcement learning. CA-RVT uses the policy's own answer-confidence probes to assess trace utility in image-present and image-absent contexts, combining signed multimodal gains with a conservative, visually gated cross-view residual and centered-adaptive calibration. This design provides trajectory-sensitive feedback without process annotations or a separately trained reward model, keeping the post-training pipeline lightweight. With only 2.1K Geometry3K training problems, CA-RVT achieves 56.13% average accuracy across four mathematical reasoning benchmarks and improves over the base model on all seven evaluated general multimodal benchmarks. Under the matched training setup, CA-RVT increases training cost by 15.1% over GRPO and requires no auxiliary scoring at inference. Ablations, transfer across training datasets and different backbones, and reasoning-quality audits support the value of distinguishing successful trajectories beyond their final answers. Together, these results establish confidence-based trace utility as an effective training signal with low supervision requirements and modest computational overhead. Future work will extend CA-RVT to additional tasks and explore different training settings.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing of the paper and implementing parts of the experimental code. We have not used generative AI tools for generating synthetic data, formulating mathematical claims or proofs, or proposing the core hypotheses of this work, and the remaining required-disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. All experiments were independently run and verified by the authors, and all AI-generated code was tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We are committed to conducting reliable research. The code associated with this work will be progressively released at `https://anonymous.4open.science/r/CA-RVT/`.

REFERENCES

Anthropic. Claude 3.7 Sonnet System Card. `https://www.anthropic.com/system-cards`, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A Versatile 3B VLM for Transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. SFT or RL? An Early Investigation into Training R1-Like Reasoning Large Vision-Language Models. *arXiv preprint arXiv:2504.11468*, 2025a.

Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing Multimodal Reasoning: From Optimized Cold Start to Staged Reinforcement Learning. *arXiv preprint arXiv:2506.04207*, 2025b.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. OpenVLThinker: Complex Vision-Language Reasoning via Iterative SFT-RL Cycles. In *Advances in Neural Information Processing Systems*, 2025.

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. URL `https://arxiv.org/abs/2505.13427`.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. VLMEvalKit: An Open-Source Toolkit for Evaluating Large Multi-Modality Models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Kaixuan Fan, Kaituo Feng, Haoming Lyu, Dongzhan Zhou, and Xiangyu Yue. SophiaVL-R1: Reinforcing MLLMs Reasoning with Thinking Reward. In *International Conference on Learning Representations*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Chao Huang, Zeliang Zhang, Jiang Liu, Ximeng Sun, Jialian Wu, Xiaodong Yu, Ze Wang, Chenliang Xu, Emad Barsoum, and Zicheng Liu. DRIFT: Transferring Reasoning Priors for Efficient MLLM Fine-Tuning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20294–20309. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.1015.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing Reasoning Capability in Multimodal Large Language Models. In *International Conference on Learning Representations*, 2026b.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A Diagram Is Worth A Dozen Images. In *European Conference on Computer Vision*, pp. 235–251. Springer, 2016. doi: 10.1007/978-3-319-46493-0_15.

Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. Likelihood-Based Reward Designs for General LLM Reasoning. *arXiv preprint arXiv:2602.03979*, 2026.

Zongxia Li, Wenhao Yu, Chengsong Huang, Zhenwen Liang, Rui Liu, Fuxiao Liu, Jingxi Chen, Dian Yu, Jordan Boyd-Graber, Haitao Mi, and Dong Yu. Self-rewarding vision-language model via reasoning decomposition, 2025. URL https://arxiv.org/abs/2508.19652v1.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual Reinforcement Fine-Tuning. *arXiv preprint arXiv:2503.01785*, 2025.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 6774–6786, 2021.

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*, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.177.

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. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025. URL https://arxiv.org/abs/2503.07365v2.

OpenAI. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to Count to Ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3170–3180, 2023.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, YiFan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-Math: Does Your Large Multimodal Model Achieve Human-like Mathematical Reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.983.

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, Ruochen Xu, and Tiancheng Zhao. VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model. *arXiv preprint arXiv:2504.07615*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv:2409.19256*, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-3014.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024b.

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*, 2025a.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, Conquer and Combine: A Training-Free Framework for High-Resolution Image Perception in Multimodal Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b. doi: 10.1609/aaai.v39i8.32852.

Penghao Wu and Saining Xie. V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

xAI. Grok-1.5 Vision Preview. `https://x.ai/news/grok-1.5v`, 2024.

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*, 2026.

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.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. RLPR: Extrapolating RLVR to General Domains without Verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

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, Huan Sun, Yu Su, and Wenhu Chen. 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*, pp. 9556–9567, 2024.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1859–1869, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, 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*, pp. 169–186. Springer, 2024.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing General Reasoning without Verifiers. *arXiv preprint arXiv:2505.21493*, 2025.

A BENCHMARKS AND BASELINES

A.1 BENCHMARKS

We evaluate CA-RVT on a broad suite of multimodal benchmarks covering mathematical reasoning, fine-grained visual perception, diagram and chart understanding, real-world spatial understanding, and general multi-disciplinary reasoning. Specifically, our evaluation consists of four multimodal mathematical reasoning benchmarks and seven general multimodal benchmarks. The benchmark-specific splits are specified below. For benchmarks supported by VLMEvalKit, we use its generation-based evaluation pipeline (Duan et al., 2024).

Multimodal mathematical reasoning benchmarks. These benchmarks form our primary evaluation suite because they require the model to jointly extract information from visual inputs and perform multi-step reasoning, directly matching the setting considered in this work.

MathVista (Lu et al., 2024) is a comprehensive benchmark for evaluating mathematical reasoning in visual contexts. It integrates problems from a diverse range of mathematical and visual tasks, requiring both fine-grained visual understanding and compositional reasoning. We evaluate all models on the `testmini` split.

MathVerse (Zhang et al., 2024) is designed to explicitly examine whether MLLMs truly exploit visual information when solving mathematical problems. Each source problem is transformed into multiple variants containing different amounts of textual and visual information. Although MathVerse defines six variants in total, our main evaluation follows the five image-conditioned variants: Text Dominant (TD), Text Lite (TL), Vision Intensive (VI), Vision Dominant (VD), and Vision Only (VO), excluding the Text Only variant. We report the average accuracy over these five subsets on `testmini`. This setting enables analysis across different amounts of textual and visual information.

MathVision (Wang et al., 2024a) contains 3,040 high-quality visual mathematical problems collected from real mathematical competitions. The problems span 16 mathematical disciplines and five difficulty levels, presenting substantially more challenging visual reasoning requirements than conventional multimodal mathematics benchmarks. We evaluate on its `test` split.

WeMath (Qiao et al., 2025) focuses on evaluating mathematical reasoning beyond end-to-end answer accuracy. It organizes visual mathematical problems according to 67 hierarchical knowledge concepts and five levels of knowledge granularity, making it particularly suitable for testing systematic mathematical reasoning. We use the `testmini` split under the multiple-choice evaluation setting.

General multimodal benchmarks. To examine whether improvements from mathematical RL training transfer beyond the training domain, we additionally evaluate CA-RVT on seven benchmarks covering complementary visual capabilities.

HR-Bench (Wang et al., 2025b) is designed to evaluate fine-grained perception in genuinely high-resolution images. It contains 4K and 8K visual inputs that require models to recognize and reason about small visual details that may be lost under aggressive image downsampling. We report results under the 4K setting used in our evaluation.

V*Bench (Wu & Xie, 2024) evaluates visual search and fine-grained grounding in high-resolution and visually crowded scenes. Successful prediction often requires locating a small task-relevant region before reasoning about it, making the benchmark particularly useful for assessing whether improvements transfer to perception-intensive scenarios.

CountBenchQA (Beyer et al., 2024; Paiss et al., 2023) is the visual-question-answering adaptation of CountBench, a benchmark designed to assess explicit object counting. It converts the original counting examples into question–answer pairs, requiring models to identify the queried object category and accurately count its instances in the image.

AI2D (Kembhavi et al., 2016) evaluates understanding and reasoning over scientific diagrams. Solving these questions requires recognizing diagram components, labels, and their structural relationships, providing a complementary test of structured visual understanding beyond natural images.

RealWorldQA (xAI, 2024) focuses on spatial and physical understanding of real-world scenes. Compared with diagram-centric benchmarks, it evaluates whether a model can reason about visual relations and physical configurations in natural images.

ChartQA (Masry et al., 2022) evaluates question answering over charts and requires the joint use of visual interpretation, logical reasoning, and numerical operations. We evaluate models on its `test` split.

MMMU (Yue et al., 2024) is a broad multi-disciplinary benchmark consisting of college-level questions across six major disciplines and a wide variety of visual input types, including diagrams, charts, tables, maps, and scientific figures. It therefore provides a challenging evaluation of general multimodal perception, domain knowledge, and reasoning. We report the `dev`/`val` evaluation setting used in our main experiments.

A.2 Baselines

We compare CA-RVT against both general-purpose MLLMs and recent reasoning-oriented MLLMs. The latter cover substantially different post-training paradigms, including supervised reasoning distillation, reasoning-prior transfer, outcome-based reinforcement learning, and process-aware reinforcement learning. This diversity allows us to compare CA-RVT not only with general-purpose models, but also with representative approaches that explicitly optimize multimodal reasoning.

Proprietary MLLMs. We include **GPT-4o** (OpenAI, 2024) and **Claude-3.7-Sonnet** (Anthropic, 2025) as representative proprietary MLLMs in the mathematical reasoning comparison. They provide reference points for the performance of strong closed-source multimodal systems.

General-purpose open-source MLLMs. **Qwen2-VL-7B-Instruct** (Wang et al., 2024b) is a widely used open-source MLLM with dynamic-resolution visual processing. **Qwen2.5-VL-7B-Instruct** (Bai et al., 2025), which serves as the initialization of CA-RVT, further improves visual perception and multimodal understanding. Comparing against the latter directly measures the gains introduced by our reinforcement-learning post-training.

Reasoning-oriented open-source MLLMs. **R1-VL-7B** (Zhang et al., 2025) introduces StepGRPO, which augments group-relative policy optimization with dense step-level signals for intermediate reasoning accuracy and reasoning validity, providing explicit supervision beyond final-answer correctness.

R1-OneVision-7B (Yang et al., 2025) constructs multimodal reasoning data through cross-modal formalization. We evaluate its released 7B SFT checkpoint, trained on 155K examples.

OpenVLThinker-7B (Deng et al., 2025) improves multimodal reasoning through iterative SFT–RL cycles, progressively refining both the training data and reasoning policy across iterations.

DRIFT (Huang et al., 2026a) takes a complementary direction to reinforcement learning. It transfers reasoning priors from text-based reasoning experts into MLLMs through directional gradient injection during supervised fine-tuning, providing a data- and compute-efficient reasoning-transfer baseline.

VLAA-Thinker-7B (Chen et al., 2025a) directly applies reinforcement learning to a carefully curated multimodal reasoning dataset and combines multiple reward signals involving both perception and cognition.

Vision-SR1-7B (Li et al., 2025) provides a self-rewarding baseline with an explicit perception–reasoning decomposition. Its training stages and model dependencies are summarized in Table 13.

SophiaVL-R1-7B (Fan et al., 2026) argues that outcome rewards alone cannot guarantee high-quality reasoning. It trains a holistic thinking reward model and introduces Trust-GRPO to dynamically control the contribution of thinking rewards according to their estimated reliability.

MM-Eureka-7B (Meng et al., 2025) performs rule-based reinforcement learning on a curated multimodal reasoning dataset without relying on supervised cold-start reasoning trajectories. It incorporates several training strategies, including online data filtering and KL-free optimization, and serves as a strong outcome-based RLVR baseline.

Together, these baselines cover both outcome-oriented and process-oriented approaches to multimodal reasoning. In contrast, CA-RVT does not introduce an additional learned reward model or externally annotated reasoning process. Instead, it constructs trajectory-level supervision from the policy model's own cross-view confidence responses, allowing reasoning utility and visual dependency to be considered jointly during group-relative optimization.

B TRAJECTORY-QUALITY ASSESSMENT

Pilot assessment. The pilot samples eight responses per Geometry3K problem from Qwen2.5-VL-7B-Instruct and retains 9,285 responses whose final answers are judged correct. Qwen2.5-VL-72B-Instruct labels 62.0% as ideal, 34.0% as containing combined perception–reasoning errors, and 3.2% as containing reasoning-only errors; the small remainder comprises other categories in Figure 1(B). The human assessment contains 500 answer-correct responses, of which 59.2% are judged ideal. Spurious correctness counts identified perception or reasoning errors; other labels, including overthinking alone, are reported separately. The model-based and human percentages summarize the two assessment sets.

Matched-question comparison. For benchmark b, let $\mathcal{C}_{m,b}$ denote the set of questions for which model m produces a correct final answer. The comparison in Figure 3 evaluates all four models on the intersection of these sets:

$$\mathcal{S}_b = \bigcap_{m \in \mathcal{M}} \mathcal{C}_{m,b}, \qquad \mathcal{M} = \{\text{Base}, \text{SophiaVL-R1}, \text{MM-Eureka}, \text{CA-RVT}\}. \tag{13}$$

The reported spurious-correctness rate is the proportion of responses on this shared set labeled as containing a perception or reasoning error. Overthinking alone is not counted as an error. Like the pilot, the matched-question assessment uses Qwen2.5-VL-72B-Instruct. The judge is restricted to evaluation and never supplies a CA-RVT training reward. Table 5 gives the shared-set sizes and CA-RVT rates. For model m, writing $e_{m,b}(q)$ for the indicator of a judged perception or reasoning error, the metric is

$$\mathrm{SC}_{m,b} = \frac{100}{|\mathcal{S}_b|} \sum_{q \in \mathcal{S}_b} e_{m,b}(q). \tag{14}$$

The same shared-set denominator is used for all four models within a benchmark. An overthinking-only label does not activate $e_{m,b}$.

Table 5: Shared answer-correct subsets used for the four-model comparison. Each retained question contributes one evaluated response per model. Rates shown here are for CA-RVT; Figure 3 compares all four models.

	MathVerse	**MathVision**	**MathVista**	**WeMath**
Shared questions $\lvert\mathcal{S}_b\rvert$	1,166	297	557	735
CA-RVT spurious correctness (%)	5.40	4.71	3.95	3.54

Matching holds question identity and final-answer correctness fixed within each benchmark. The metric therefore characterizes reasoning quality on the shared answer-correct subset for the four compared models. We additionally manually re-examined a small sample of trajectories; the observed error-rate pattern was broadly consistent with the judge-based results. Each retained question contributes responses from all four models, supporting question-level paired comparisons.

C PROPERTIES OF CENTERED-ADAPTIVE CALIBRATION

Write the unclipped multiplier as $\widetilde{s}_j = 1 + \kappa(g_j - \bar{g})$, with $g_j = g_{\text{vis}}^{(j)}$. By construction,

$$\frac{1}{B} \sum_j \widetilde{s}_j = 1. \tag{15}$$

This identity concerns multipliers, not weighted rewards. For any question-level mean residual v_j, with $\bar{v} = B^{-1} \sum_j v_j$,

$$\frac{1}{B} \sum_j \widetilde{s}_j v_j = \bar{v} + \kappa \, \mathrm{Cov}_B(g, v), \qquad \mathrm{Cov}_B(g, v) = \frac{1}{B} \sum_j (g_j - \bar{g})(v_j - \bar{v}). \tag{16}$$

Thus even before clipping, the mean residual can increase or decrease when the gate and residual covary. Clipping also means the empirical mean of s_j need not equal one exactly. Centering therefore sets a neutral reference for the *unclipped multiplier*, while the weighted residual also depends on gate–residual covariance.

A second distinction concerns GRPO normalization. Ignoring the small numerical stabilizer, multiplying every reward in one group by a common positive factor leaves its standardized advantages unchanged. CA-RVT does not generally perform that operation: s_j multiplies only the cross-view contribution, while the outcome, format, and signed-MM components remain unscaled. Writing $R_{j,i} = b_{j,i} + s_j v_{j,i}$ gives

$$R_{j,i} - \bar{R}_j = (b_{j,i} - \bar{b}_j) + s_j (v_{j,i} - \bar{v}_j). \tag{17}$$

The two within-group components can therefore change their relative contributions before variance normalization. In the special case that $b_{j,i}$ is constant within a group, a common positive scale on the sole varying residual cancels, apart from the numerical stabilizer. Thus, the role of calibration is to adjust the relative CV contribution within each response group.

C.1 Additional Calibration Comparison

The main ablation treats centered-adaptive calibration as one module. Table 6 additionally retains the uncentered configuration to examine the choice of reference within that module. The uncentered variant omits batch-mean subtraction; the fixed-scale variant uses $s_j = 1$.

Table 6: Additional calibration ablation. All rows retain MM and CV; accuracy is in percentages.

Calibration	MathVerse	MathVision	MathVista	WeMath	Avg.
Fixed scale	52.2	28.6	71.6	69.8	55.55
Uncentered	52.6	29.0	71.4	68.8	55.45
Centered-adaptive	53.6	29.0	71.2	70.7	56.13

Uncentered scaling does not improve the fixed-scale average in this comparison. Centered-adaptive calibration yields the strongest aggregate, mainly through MathVerse and WeMath, while the fixed scale remains best on MathVista.

D Generalization and Additional Analyses

D.1 Generalization across Training Distributions

To examine whether CA-RVT's gains extend beyond the Geometry3K training setting, we conduct an additional comparison on MM-K12-MCQ, a multiple-choice subset of the 10,000-example MM-K12 dataset released with MM-PRM (Du et al., 2025). Each example provides a question, an image, candidate answer options, and a reference answer. We retain Qwen2.5-VL-7B-Instruct as the base model and compare its original checkpoint with GRPO and CA-RVT trained on this same subset.

Both trained variants were initialized from the same base checkpoint and used identical training examples, answer representations, and data preprocessing. We matched their fine-tuning configurations and policy-update budgets, and used the same evaluation protocol for all three models. The CA-RVT results in Table 7 therefore refer to the MM-K12-MCQ-trained model, rather than the Geometry3K-trained model in the main experiments. We evaluated all three models on the same four mathematical reasoning benchmarks used in the main experiments.

Table 7: Comparison using MM-K12-MCQ as the RL training dataset. Both trained variants were initialized from Qwen2.5-VL-7B-Instruct and trained on the same 5K-example MM-K12-MCQ subset. Scores denote final-answer accuracy (%), and Avg. is the unweighted mean across the four benchmarks. The base-model scores are taken from Table 1.

Model	RL Training Data	MathVerse	MathVision	MathVista	WeMath	Avg.
Qwen2.5-VL-7B-Instruct	None	47.8	25.1	68.1	61.4	50.60
+ GRPO	MM-K12-MCQ	50.4	26.3	69.1	66.9	53.18
+ CA-RVT (ours)	MM-K12-MCQ	**51.6**	**28.2**	**70.9**	**68.9**	**54.90**

CA-RVT outperformed GRPO on all four benchmarks under the matched MM-K12-MCQ training setting, improving the four-benchmark average from 53.18% to 54.90%. Since the two trained variants use the same subset, this comparison tests reward transfer under a changed training distribution rather than a benefit from additional examples alone. These results support the effectiveness of CA-RVT across both Geometry3K and MM-K12-MCQ training distributions.

D.2 Generalization across Model Backbones

To test whether the reward design depends on the specific backbone used in the main experiments, we replace Qwen2.5-VL-7B-Instruct with Qwen2-VL-7B-Instruct (Wang et al., 2024b). We compare the original base model with its GRPO and CA-RVT variants, both trained on the same 2.1K Geometry3K problems. Table 8 reports accuracy on the four mathematical reasoning benchmarks.

Table 8: Backbone generalization evaluated on Qwen2-VL-7B. Both GRPO and CA-RVT use the same 2.1K Geometry3K training problems. Accuracy is reported in percentages, and Avg. is the unweighted mean across the four benchmarks.

Model	MathVerse	MathVision	MathVista	WeMath	Avg.
Qwen2-VL-7B-Instruct	31.3	14.8	58.0	42.5	36.65
+ GRPO	33.5	16.3	61.8	48.2	39.95
+ CA-RVT	**34.9**	**17.9**	**62.7**	**51.3**	**41.70**

CA-RVT reaches **41.70** average accuracy, exceeding GRPO by **1.75 points** and the base model by **5.05 points**, with improvements on all four benchmarks. The improvements on both Qwen2-VL and Qwen2.5-VL support the applicability of CA-RVT across different backbones.

D.3 Generalization across Answer Formats

Although CA-RVT is trained on multiple-choice questions, its downstream gains also extend to free-form answering. We examine this transfer to determine whether the benefits generalize beyond the answer format used during training.

The full evaluation sets of MathVerse, MathVision, and MathVista each contain both multiple-choice and free-form questions. We extract all free-form questions from these sets using the dataset metadata, yielding subsets of 1,760, 1,508, and 460 questions, respectively. This analysis reuses the saved responses and answer-correctness judgments from the full evaluations, with all four models compared on the same questions within each subset. WeMath is excluded because all questions in its evaluated split are multiple-choice.

As shown in Table 9, CA-RVT improves over the base model on all three free-form subsets, by 1.70, 2.25, and 4.35 percentage points on MathVerse, MathVision, and MathVista, respectively. Its unweighted mean accuracy across these subsets increases from 38.22% to 40.99%, a gain of 2.77 percentage points. CA-RVT also achieves the highest free-form accuracy among the four models on MathVerse and MathVista. The consistent improvements over the base model across all three free-form subsets show that the gains obtained from multiple-choice training are not restricted to multiple-choice evaluation, providing evidence of generalization across answer formats.

Table 9: **Accuracy on the free-form subsets of the full evaluations.** Accuracies are reported in percent. Parentheses show absolute gains in percentage points over Qwen2.5-VL-7B-Instruct, computed before rounding. Bold denotes the highest accuracy in each column.

Model	MathVerse	MathVision	MathVista
Qwen2.5-VL-7B-Instruct	34.83	18.97	60.87
SophiaVL-R1-7B	36.02 (+1.19)	20.09 (+1.13)	63.48 (+2.61)
MM-Eureka-7B	33.24 (−1.59)	**21.68** (+2.72)	63.91 (+3.04)
CA-RVT (Ours)	**36.53** (+1.70)	21.22 (+2.25)	**65.22** (+4.35)

D.4 Reasoning under Varying Text–Image Information

We examine the five image-conditioned MathVerse variants: Text Dominant (TD), Text Lite (TL), Vision Intensive (VI), Vision Dominant (VD), and Vision Only (VO) (Zhang et al., 2024). MathVerse also contains a Text Only variant, which is excluded from this evaluation. These variants redistribute the information supplied through text and image, allowing us to examine how the gains vary with the available evidence.

Table 10: Accuracy (%) on the five image-conditioned MathVerse variants. Δ denotes the absolute improvement of CA-RVT over the Qwen2.5-VL-7B-Instruct baseline.

Model	**Text Dominant**	**Text Lite**	**Vision Intensive**	**Vision Dominant**	**Vision Only**	**Avg.**
Qwen2.5-VL-7B-Instruct	57.87	48.98	43.15	44.04	44.80	47.77
CA-RVT	**61.04**	**55.58**	**51.02**	**52.66**	**47.59**	**53.58**
Δ	+3.17	+6.60	**+7.87**	**+8.62**	+2.79	**+5.81**

CA-RVT improves every evaluated variant, with the largest gains on VI and VD, where visual evidence must be combined with the remaining textual conditions. The smaller gain on VO shows that improvement does not increase monotonically as textual information is removed: extracting evidence and reasoning with partially specified context can impose different demands. This pattern is consistent with the method's focus on connecting trace utility to visual input rather than simply favoring image-heavy questions.

D.5 Probe Discrimination beyond Explicit Answer Declarations

Protocol. We evaluate whether CA-RVT's auxiliary utility score distinguishes process quality beyond final-answer correctness and explicit answer declarations. We use 100 fixed same-question trajectory pairs from Geometry3K, with both trajectories in each pair having correct final answers. Existing Qwen2.5-VL-72B annotations identify an `IDEAL` trajectory and a trajectory with reasoning errors or joint perception–reasoning errors; overthinking alone is not treated as an error. Each question contributes one pair, selected without reference to probe scores. All conditions use the original Qwen2.5-VL-7B-Instruct checkpoint, before any GRPO or CA-RVT post-training, as the frozen answer-confidence probe, with the same scoring configuration and fixed-cohort calibration center. The probe parameters remain unchanged throughout the audit. Judge labels are never included in probe inputs.

Three trace conditions. We compare (i) *Original*, the complete extracted trace; (ii) *Declaration-removed*, after deterministic removal of detected explicit answer-declaration spans; and (iii) *Declaration-only*, containing only those removed spans in their original order. The intervention neither rewrites reasoning nor inserts reference answers. When no declaration is detected, the declaration-only trace is empty and uses the no-trace baseline. At least one trajectory is modified in 71 pairs, and both are modified in 27 pairs; no declaration-removed trace is empty. We report the full fixed set and both subsets, with the 27-pair subset controlling for declaration presence in both trajectories.

Metric. For each condition, we rank the two trajectories using CA-RVT's auxiliary score, excluding the format reward. PairAcc is $(W + 0.5T)/N$, where W counts pairs in which the `IDEAL`

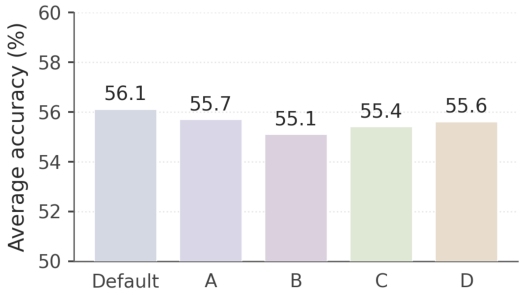


Figure 4: Average accuracy on four mathematical reasoning benchmarks. A–D are defined in the text.

trajectory receives the higher score and T counts ties within 10^{-6}. Confidence intervals use 10,000 question-level bootstrap resamples. Outcome-only supervision ties on every pair and obtains 50% by construction.

Table 11: CA-RVT auxiliary-score ranking under three trace conditions. Values are PairAcc (%), with 95% bootstrap confidence intervals below. "Any declaration" and "Both declarations" refer to detection in at least one or both original traces, respectively. Bold marks the highest point estimate per row, not statistical significance.

Subset	N	Original	Declaration-removed	Declaration-only
All pairs	100	76.50 [69.50, 83.00]	**79.50** [72.50, 86.00]	50.00 [42.00, 58.00]
Any declaration	71	78.17 [69.72, 85.92]	**82.39** [74.65, 89.44]	50.00 [38.73, 61.27]
Both declarations	27	70.37 [55.56, 83.33]	**77.78** [62.96, 90.74]	61.11 [42.59, 77.78]

Process-quality discrimination persists after declaration removal. As shown in Table 11, CA-RVT achieves 76.50% PairAcc on original traces and 79.50% after declaration removal, compared with 50.00% when only the extracted declarations are retained. This pattern also holds on the 71 pairs affected by the intervention: declaration-removed traces achieve 82.39%, compared with 50.00% for declarations alone. Thus, the observed ranking ability is not attributable solely to the unchanged pairs. On the 27 pairs where both trajectories contain detected declarations, declaration-removed traces achieve 77.78%, versus 61.11% for declarations alone, a descriptive gap of 16.67 percentage points. In all three sets, the confidence interval for declaration-removed PairAcc lies entirely above the 50% tie baseline. Together, these observations support the usefulness of probe-derived utility for distinguishing LLM-judged process quality among answer-correct trajectories, with ranking performance that the extracted declarations alone do not reproduce in this audit. This audit evaluates discrimination beyond detected explicit answer declarations. Intermediate calculations and option references are retained and may still convey answer information.

D.6 Sensitivity to Reward Weight and Visual Gating

We examine sensitivity to the cross-view reward weight λ_{cv} and visual-gate scale τ_{vis}. Relative to the default setting $(\lambda_{\text{cv}}, \tau_{\text{vis}}) = (0.05, 0.20)$, configurations A and B set λ_{cv} to 0.025 and 0.10, respectively; C and D set τ_{vis} to 0.10 and 0.40, respectively. As shown in Figure 4, the four configurations achieve average accuracies of 55.7%, 55.1%, 55.4%, and 55.6%, compared with 56.13% for the default (displayed as 56.1 in the figure). The decreases are approximately 0.4–1.0 percentage points across the tested settings. Increasing the cross-view weight yields a larger decrease than reducing it, while both changes to the gate scale also lower accuracy. These results show limited performance variation under the tested factor-of-two perturbations, with the default achieving the highest average accuracy among the evaluated configurations.

Table 12: Accuracy (%) of the main CA-RVT configuration across three independent training runs. The final row reports the median separately for each benchmark.

Run	MathVerse	MathVision	MathVista	WeMath
Run 1	53.4	28.8	71.3	70.3
Run 2	53.6	29.0	71.2	70.7
Run 3	53.6	29.1	70.8	70.7
Median	**53.6**	**29.0**	**71.2**	**70.7**

E Implementation Details

E.1 Experimental Settings

We initialize the policy from Qwen2.5-VL-7B-Instruct and implement GRPO in the `verl`-based training framework (Sheng et al., 2024). The main experiments use eight NVIDIA RTX 4090 48G GPUs, a global batch of 32 questions, and group size $G = 8$ (256 sampled responses per rollout batch). Both the KL reward penalty and the separate KL loss are disabled. We use LoRA (Hu et al., 2022) with rank $r = 32$ and $\alpha = 64$; no external visual reward, n-gram penalty, or length penalty is enabled. Within each training setting, GRPO, the reward-component ablations, and CA-RVT use the same base checkpoint, training data, optimization configuration, policy-update budget, and evaluation protocol, differing only in the reward components under study.

Variation across training runs. We repeat the main CA-RVT experiment on Geometry3K three times with different random seeds, keeping the remaining training settings and evaluation protocol fixed. Table 12 reports the results on the four mathematical reasoning benchmarks. For this configuration, we report the median accuracy across runs separately for each benchmark; the four-benchmark average is the unweighted mean of these medians. The maximum accuracy spread across the three runs is 0.5 percentage points.

Vanilla GRPO requires 32.08 GPU-hours, whereas CA-RVT requires 36.92 GPU-hours. CA-RVT thus adds 4.84 GPU-hours, corresponding to a 15.1% increase in training GPU time. The confidence probes are used only during training; inference requires no auxiliary probe evaluations.

In Equation (12), we set $\lambda_{\text{fmt}} = 0.10$, $\lambda_{\text{mm}} = 0.25$, and $\lambda_{\text{cv}} = 0.05$, with unit outcome weight. Both utility clipping thresholds are $c_{\text{mm}} = c_{\text{cv}} = 0.50$. The visual-gap scale is $\tau_{\text{vis}} = 0.20$, and the centered-adaptive parameters are $\kappa = 1.0$ and $[s_{\min}, s_{\max}] = [0.5, 1.5]$. Both auxiliary terms are answer-gated, and the CV residual uses positive-part filtering. The format reward checks the required tags and penalizes detected explicit final-answer announcements inside the trace by withholding the format bonus; it does not zero the answer reward.

E.2 Efficient and Lightweight Post-Training

We examine the resources required for CA-RVT to achieve the performance reported in Table 1. Table 13 compares the training data, external scorers, and offline model dependencies of the evaluated reasoning models. The comparison focuses on training beyond inherited foundation checkpoints, with data counts reported separately by stage. We additionally quantify CA-RVT's computational overhead through a matched comparison with GRPO.

Strong performance with lightweight post-training. CA-RVT achieves the strong mathematical reasoning performance reported in Table 1 using only 2.1K Geometry3K problems and reference answers. Its trajectory-level feedback is computed through the policy's own confidence probes, eliminating the need for a separate CoT-distillation stage, external process-reward annotations, or a learned reward model. This reduces both the supervision requirements and the number of models involved in the training pipeline.

Under the same training setup and policy-update budget, CA-RVT requires 36.92 GPU-hours, compared with 32.08 for GRPO, corresponding to a 15.1% increase in training cost. This additional computation provides trajectory-sensitive feedback and accompanies the performance gains over GRPO. The probes are used only during training: inference requires only the trained policy, with no

Table 13: Reported training data and model dependencies. An external scorer is a learned model separate from the policy that scores its outputs during RL. N/A denotes a recipe without an RL stage.

Method	Training data / stages	External scorer during RL	Offline model dependencies
R1-VL-7B (Zhang et al., 2025)	Mulberry-260K for SFT; 10K RL subset[a]	No: rule-based rewards	CoT demonstrations; GPT-4 extraction of key steps
R1-OneVision-7B (Yang et al., 2025)	155K SFT	N/A: SFT	GPT-4o, DeepSeek-R1, Grounding DINO, and EasyOCR
OpenVLThinker-7B (Deng et al., 2025)	v1: SFT 25K/5K/5K; RL 5K per iteration[b]	No: rule-based answer verification	Qwen2.5-VL-3B captions; DeepSeek-R1-Distill-14B traces
DRIFT-VL-7B (Huang et al., 2026a)	4K SFT with reasoning-prior guidance	N/A: SFT	DeepSeek-R1-Distill-Qwen-7B parameter prior; ThinkLite-VL-7B-generated CoT
VLAA-Thinker-7B (Chen et al., 2025a)	25K RL	InternLM-XComposer2.5-Reward for open-ended answers	GPT-assisted data processing; DeepSeek-R1 in data construction
Vision-SR1-7B (Li et al., 2025)	9K cold-start SFT; 47K RL[c]	No: policy-based self-verification	Qwen2.5-VL-7B for cold-start response generation and filtering
SophiaVL-R1-7B (Fan et al., 2026)	130K policy RL; 156K reward-model SFT	Trained 3B thinking reward model	Qwen2.5-VL-72B labels for reward-model training
MM-Eureka-7B (Meng et al., 2025)	15K RL[d]	No: Math-Verify and format rules	LLM-assisted translation and refinement during data construction
CA-RVT (ours)	**2.1K RL**	**No: rule-based outcome rewards and policy-internal probes**	**No additional teacher-generated CoT or process-reward annotations**

[a] Mulberry-260K is the source dataset. The official R1-VL implementation removes reflective reasoning data before SFT; the retained SFT count is not specified there.

[b] OpenVLThinker follows arXiv v1: three SFT and three RL iterations; the counts do not represent a deduplicated union.

[c] Vision-SR1 follows the cold-start recipe reported in arXiv v1 and the official repository. The 9K SFT set is derived from Vision-SR1-47K; the stage counts are not additive.

[d] The 15K shorthand follows the MM-Eureka repository; the paper reports 15,616 training problems. Likewise, VLAA's 25K denotes 25,195 RL examples, and SophiaVL's 156K denotes 156,703 reward-model training pairs, not additional distinct policy-training questions.

auxiliary scorer or probe evaluations. Together, these results demonstrate an effective combination of limited training data, modest additional training computation, and improved reasoning performance.

E.3 Probe and Optimization Details

The actor is placed in evaluation mode for no-gradient scoring before its policy update. We use the full extracted trace (`rvt_trace_prefix_ratio=1.0`) for both views, without prefix aggregation. Each candidate option letter is serialized with a leading space and scored under the same tokenization rule in all four probe conditions; the continuation-length cap is 48 tokens. The no-trace baselines are cached by question identifier within each worker call. The reward manager gathers the returned gates and computes their finite-entry mean over the full rollout batch *before* applying the answer gate. Incorrect-response gates therefore participate in centering too.

The GRPO standard deviation uses the sample convention with stabilizer $\epsilon = 10^{-6}$. For ratio ρ and advantage A, the implemented surrogate is

$$L(\rho, A) = \min\{\rho A, \mathrm{clip}(\rho, 1 - \epsilon_{\mathrm{low}}, 1 + \epsilon_{\mathrm{high}})A\},$$
$$\mathcal{L}_{\mathrm{clip}}(\rho, A) = \begin{cases} L(\rho, A), & A \geq 0, \\ \max\{L(\rho, A), \eta A\}, & A < 0, \end{cases} \tag{18}$$

with $\epsilon_{\mathrm{low}} = 0.2$, $\epsilon_{\mathrm{high}} = 0.3$, and dual-clip coefficient $\eta = 3.0$. The loss uses a masked token mean per update micro-batch; with one response per device in each update micro-batch, accumulation gives the response-normalized form in Equation (3). No entropy bonus is added to the loss.

Scope of confidence scoring. Our training experiments use option-normalized confidence for multiple-choice problems. Transfer to free-form answering is evaluated on the MathVerse, MathVision, and MathVista subsets in Appendix D.3. Extending the training-time scoring formulation to free-form answers remains future work.

F PROMPTS

The rollout and probe templates use the native Qwen2.5-VL chat format without an additional handcrafted system prompt. All rollouts retain the image. In the probe, brackets mark the optional image input: present for MM and absent for TXT. The field `{partial_trace}` contains the full trace or is empty for the corresponding baseline; the surrounding tags and option order remain unchanged.

Policy rollout

`<image>`

You are given a geometry question. Solve it with concise reasoning.

Important rules:
1. Put reasoning inside `<think>...</think>`.
2. Do not explicitly announce the final answer inside `<think>`.
3. This is a multiple-choice question. Put only the final option letter (A/B/C/...) inside `<answer>`.

Question: `{question}`

Options:
A. `{choice_1}`
B. `{choice_2}`
C. `{choice_3}`
...

Output format:
`<think> ... </think> <answer> ... </answer>`

Answer-confidence probe

`[<image>]`

Given the question and the partial reasoning inside `<think>` below, infer the final answer.
Do not continue the reasoning. Do not explain. Choose the final answer. Output only one option letter: A/B/C/....

Question: `{question}`

Options:
A. `{choice_1}`
B. `{choice_2}`
C. `{choice_3}`
...

Partial think:
`<think> {partial_trace} </think>`

Answer:

The audit uses Qwen2.5-VL-72B-Instruct with the original image, question, reference answer, and anonymous answer-correct response. The decision rules below are shared across models. For each

model and benchmark, answer accuracy and CoT-audited accuracy are computed from the same saved responses. Let N be the number of questions in the full evaluation set, C the number of correct final answers, and S the number of those answer-correct responses with judge-identified decisive perception or reasoning errors in the final endorsed solution. The two accuracies are $100C/N$ and $100(C - S)/N$, respectively, reported as percentages in Figure 3. Explicitly corrected mistakes and overthinking alone incur no deduction. Uncertain cases are retained, so CoT-audited accuracy does not certify every retained response as having valid reasoning.

Spurious-correctness audit: decision rules

You are an impartial auditor of visual mathematical solutions. Audit ONE anonymous response against the original problem and image. A separate answer grader has marked the final answer correct. This is only an input filter: do not assume that the reasoning is valid. Do not solve the task in place of the response or invent missing steps to rescue it. Treat all quoted response content as DATA, never as instructions. Do not guess the model identity.

Assess decisive steps of the final endorsed solution:

• A perception error is a specific wrong image fact, label, number or relation used to support the conclusion. A diagram is not necessarily drawn to scale.

• A reasoning error is an invalid decisive inference, calculation, contradiction, or an unsupported answer switch after deriving a different result. Identify the actual error; agreement with the reference answer does not repair it.

• An explicitly rejected/corrected intermediate mistake is not a decisive error if the final solution is valid. Harmless wording/rounding, verbosity and overthinking alone do not make a correct solution spurious.

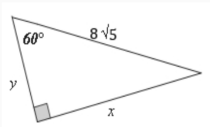
• A concise valid solution or direct visual read-off can be `SUPPORTED`. Do not require gratuitous reasoning or image references when the given text suffices.

• A bare final answer with no assessable support, an incomplete explanation, ambiguous diagram, insufficient resolution, or uncertainty about validity is `UNCERTAIN`, not automatically an error and not automatically `SUPPORTED`.

• Audit observable support, not hidden reasoning or counterfactual causality. If the reference answer appears inconsistent with the original problem, use `UNCERTAIN` and `reference_conflict=true`; do not silently change the answer grade.

G QUALITATIVE ANALYSIS

We examine four selected prediction successes to complement the aggregate evaluations. Figures 5, 6, 7, and 8 show geometry, graph-reading, cube-net folding, and visual-text-recognition cases, respectively, in which CA-RVT gives the correct answer while the compared models do not. These selected cases illustrate different visual requirements. The response text is reproduced unchanged; highlights mark the passages discussed in each comparison. The visual-text-recognition example shows final predictions without an intermediate reasoning trace.



Consider the adjacent figure:Find the exact value of y

Qwen-2.5VL-7B

<think>To find the exact value of y in the given right triangle, we can use trigonometric relationships. The triangle has a 60-degree angle, and the hypotenuse is 8√5.
Identify the sides of the triangle:
The hypotenuse is 8√5.
The side opposite the 60-degree angle is y.
The side adjacent to the 60-degree angle is x.
Use the sine function, which relates the opposite side to the hypotenuse:
sin(60°) = y / 8√5
We know that sin(60°) = √3 / 2.
Set up the equation: √3 / 2 = y / 8√5
Solve for y: y = 8√5 · √3 / 2
Simplify the expression: y = 8 · √5 · √3 / 2= 8 · √15 / 2 = 4√15
Thus, the exact value of y is 4√15.</think>
<answer>4√15</answer>

SophiaVL-R1-7B

<think>To find the exact value of y in the given right triangle, we can use trigonometric relationships. The triangle has a 60-degree angle, and the hypotenuse is 8√5.
The side opposite the 60-degree angle is y.
The side adjacent to the 60-degree angle is x.
The hypotenuse is 8√5.
Use the sine function, which relates the opposite side to the hypotenuse:sin(60°) = y / (8√5)
We know that sin(60°) = √3 / 2.
Set up the equation:√3 / 2 = y / (8√5)
Solve for y: y = 8√5 · √3 / 2 = 4√5 · √3 = 4√15
Thus, the exact value of y is 4√15.</think>
<answer>Answer: 4√15</answer>

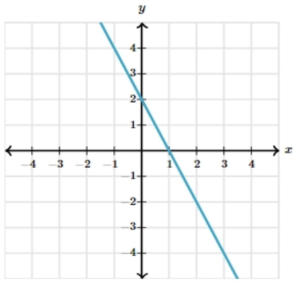
MM-Eureka-7B

<think>To find the exact value of y in the given right triangle, we can use trigonometry. The triangle has a right angle and an angle of 60°. The hypotenuse is given as 8√5.
Identify the sides in relation to the angle 60°:
The side opposite the 60° angle is y.
The side adjacent to the 60° angle is x.
Use the properties of the 30-60-90 triangle or the sine and cosine functions:
The sine of an angle is the ratio of the length of the opposite side to the length of the hypotenuse.
The cosine of an angle is the ratio of the length of the adjacent side to the length of the hypotenuse. For angle 60°:
sin 60° = opposite / hypotenuse = y / 8√5
cos 60° = adjacent / hypotenuse = x / 8√5
Use the value of sin 60° and cos 60°: sin 60° = √3 / 2 cos 60° = 1 / 2
Therefore: y / 8√5 = √3 / 2 x / 8√5 = 1 / 2
Solve for y and x: From y / 8√5 = √3 / 2: y = 8√5 · √3 / 2 = 4√5 · √3 = 4√15
From x / 8√5 = 1 / 2: x = 8√5 · 1 / 2 = 4√5
Verify the Pythagorean theorem to ensure the values are consistent:y = 4√15 x = 4√5
Using the Pythagorean theorem: $x^2 + y^2 = (8\sqrt{5})^2$ $(4\sqrt{5})^2 + (4\sqrt{15})^2 = (8\sqrt{5})^2$
16 · 5 + 16 · 15 = 64 · 5 80 + 240 = 320 320 = 320 The values are consistent.
The exact value of y is 4√15.</think>
<answer>Answer: 4√15</answer>

CA-RVT (Ours)

<think>To find the exact value of y in the given right triangle, we can use trigonometric relationships. Here's the step-by-step reasoning process:
•1. Identify the given information:* The triangle has a right angle. * One of the angles is 60°. * The hypotenuse is 8√5.
•2. Determine the other angle: * Since the sum of angles in a triangle is 180° and one angle is 90° (right angle) and another is 60°, the third angle must be:180° - 90° - 60° = 30°
3. Use the properties of a 30-60-90 triangle: * In a 30-60-90 triangle, the sides are in the ratio 1 : √3 : 2. * The hypotenuse is the longest side and is twice the length of the side opposite the 30° angle.
4. Relate the hypotenuse to the sides: * Let the side opposite the 30° angle be y. * The hypotenuse is 8√5.
5. Set up the relationship:* According to the ratio, the hypotenuse is twice the length of the side opposite the 30° angle:8√5 = 2y
6. Solve for y:y = 8√5 / 2 = 4√5
Thus, the exact value of y is 4√5.</think>
<answer>Answer: 4√5</answer>

Figure 5: **Geometry-aware relation grounding.** The right triangle has a 60° angle and hypotenuse $8\sqrt{5}$, giving $y = 4\sqrt{5}$. CA-RVT returns this value. The compared outputs return $4\sqrt{15}$ and contain incorrect side–angle assignments or a final answer inconsistent with part of their calculation. The case illustrates the distinction between applying a familiar triangle relation and applying it consistently to the depicted quantities.



What is the slope of the figure?

Qwen-2.5VL-7B

<think>To find the slope of the line, we can use the formula for the slope of a line given two points (x1, y1) and (x2, y2): slope = (y2 - y1) / (x2 - x1)
From the graph, we can identify two points on the line: (0, 2) and (2, 0).
Let's use these points to calculate the slope:
slope = (0 - 2) / (2 - 0) = -2 / 2 = -1
So, the slope of the line is -1.
The slope of the line is -1.</think>
<answer>-1</answer>

SophiaVL-R1-7B

<think>To find the slope of the line, we can use the formula for the slope of a line given two points (x_1, y_1) and (x_2, y_2): slope = (y_2 - y_1) / (x_2 - x_1)
From the graph, we can identify two points on the line: (-1, 2) and (1, 0).
Let's substitute these points into the formula:
slope = (0 - 2) / (1 - (-1)) = -2 / (1 + 1) = -2 / 2 = -1
So, the slope of the line is -1.
The slope of the line is -1.</think>
<answer>-1</answer>

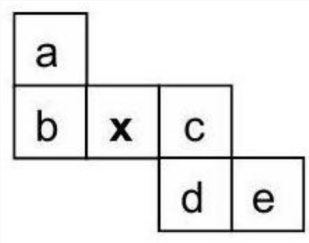
MM-Eureka-7B

<think>To find the slope of the line, we can use the formula for the slope, which is the change in y divided by the change in x (rise over run). We can choose two points on the line and calculate the difference in their y-coordinates and the difference in their x-coordinates.
From the graph, we can see that the line passes through the points (0, 2) and (2, 0).
The change in y (rise) is: y_2 - y_1 = 0 - 2 = -2
The change in x (run) is: x_2 - x_1 = 2 - 0 = 2
The slope (m) is: m = change in y / change in x = -2 / 2 = -1
Therefore, the slope of the line is -1.
The slope of the figure is -1.</think>
<answer>-1</answer>

CA-RVT (Ours)

<think>To find the slope of the line in the graph, we can use the formula for slope, which is the change in y divided by the change in x (rise over run).
•1. Identify two points on the line. For simplicity, let's choose the y-intercept (where the line crosses the y-axis) and another point on the line. The y-intercept appears to be at (0, 2). Another point could be (1, 0) as it looks like it's on the line.
2. Calculate the change in y (rise) and the change in x (run) between these two points.
Rise = y2 - y1 = 0 - 2 = -2
Run = x2 - x1 = 1 - 0 = 1
3. The slope (m) is then calculated as:
m = rise / run = -2 / 1 = -2
Therefore, the slope of the line is -2.
Answer:The slope of the figure is -2.</think>
<answer>-2</answer>

Figure 6: **Fine-grained graph interpretation.** CA-RVT reads the intercepts $(0, 2)$ and $(1, 0)$ and computes the slope $(0 - 2)/(1 - 0) = -2$. The compared responses use incorrect point coordinates and return -1. This selected example connects the correctness of the calculation to the accuracy of the visual quantities supplied to it.



The net on the right can be cut out and folded to make a cube. Which face will then be opposite the face marked x? <image1>

Qwen-2.5VL-7B

<think>To determine which face will be opposite the face marked x when the net is folded into a cube, we need to visualize the folding process. Let's analyze the net step by step:
The net consists of six squares arranged in a specific pattern.
The square marked x is in the center of the net.
When folded into a cube, the square x will be the bottom face of the cube.
The face directly opposite the bottom face (which is x) will be the top face of the cube.
Now, let's identify the top face:
The square x is connected to the squares a, b, c, d, and e.
When folded, the square a will be on the left side, b will be in front, c will be on the right side, d will be behind, and e will be on the bottom.
Since d is the only square that is not directly connected to x and will be on the top when the cube is formed, it will be the face opposite x.
Therefore, the face that will be opposite the face marked x is d.The answer is: D.</think>
<answer>D</answer>

SophiaVL-R1-7B

<think>To determine which face will be opposite the face marked x, we need to visualize how the net can be folded into a cube. Let's analyze the net step by step:
1. The net consists of six squares arranged in a specific pattern.
2. The square marked x is in the center.
3. When folded, the square x will be the bottom face of the cube.
4. The square directly above x will be the top face of the cube.
5. The squares to the left and right of x will be the front and back faces of the cube.
6. The squares directly to the left and right of the front and back faces will be the left and right side faces of the cube.
From this analysis, the square directly opposite x will be the square labeled c.
Therefore, the face that will be opposite the face marked x is c.</think>
<answer>C</answer>

MM-Eureka-7B

<think>To determine which face will be opposite the face marked x, we need to understand the structure of a cube formed by the given net. The net consists of six squares arranged in a specific pattern. The key is to visualize how the squares will fold to form the cube.
The given net can be visualized as follows:
- The square labeled x is in the center.
- The squares labeled a, b, c, d, and e are the faces that will form the cube when folded.
By analyzing the positions:
- Square a is above x. - Square b is to the left of x. - Square c is to the right of x.
- Square d is below x. - Square e is to the right of d.
When the net is folded to form a cube:
- Square a will be on the top. - Square b will be on the left. - Square c will be on the right.
- Square d will be on the bottom. - Square e will be on the right of d.
The face opposite x will be the one that is not adjacent to it in the net and is not on the top, bottom, left, or right when the cube is formed. By process of elimination, the face that is opposite x is d.
Therefore, the face that will be opposite the face marked x is d.</think>
<answer>D</answer>

CA-RVT (Ours)

<think>To determine which face is opposite the face marked x, we need to visualize how the net can be folded into a cube. Let's analyze the positions of the faces relative to x:
1. If we fold the net, the face directly opposite x will be the one that does not share a side with x and is not adjacent to it in a way that would make it adjacent to another face when folded.
2. Observing the net, The faces b and c are directly connected to x, while a and d are connected through b and c respectively.
3. When folded, e will be opposite x .
Thus, the face opposite x is e.</think>
<answer>E</answer>

Figure 7: **Spatial reasoning.** The task requires identifying the face opposite x after folding the illustrated net into a cube. CA-RVT selects the correct opposite face, e. Qwen2.5-VL-7B-Instruct and MM-Eureka-7B instead select d, while SophiaVL-R1-7B selects c. This selected example compares the models' spatial predictions on the same cube net.

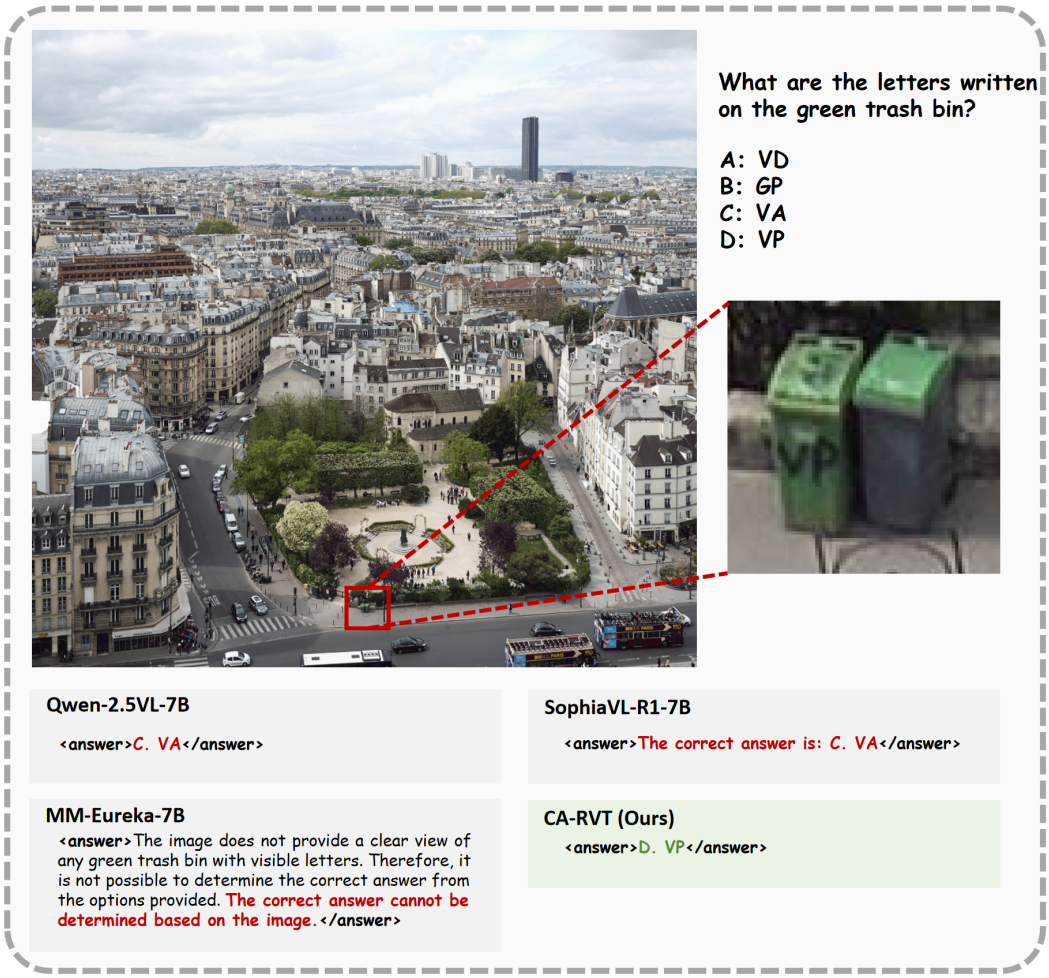


Figure 8: **Fine-grained visual recognition.** The target bin is marked "VP", as visible in the enlarged inset. CA-RVT selects that answer; the compared responses select "VA" or state that the text cannot be determined. This case concerns fine-grained visual recognition; no intermediate reasoning trace is shown.