WHEN VLMS RELY LESS ON THEIR REASONING: ADAPTIVE KL REGULARIZATION FOR RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has shown promise for improving reasoning in vision-language models (VLMs). Productive exploration is important for continued performance gains, motivating efforts to relax reference-policy constraints. However, removing reference-policy KL regularization in VLM RLVR can degrade reasoning–answer consistency even when final-answer accuracy remains comparable. Our analyses suggest that RLVR reduces VLMs' reliance on generated reasoning relative to visual inputs during answer prediction. To address this issue, we propose PEAK (Position- and Entropy-Aware KL Anchoring), a token-adaptive KL regularization method that uses the pre-RLVR model as a reference to help preserve reliance on generated reasoning while encouraging productive exploration. PEAK is guided by two observations: (1) reasoning–answer inconsistent responses exhibit larger actor–reference entropy gaps, particularly without KL regularization; so PEAK assigns stronger regularization to tokens with larger entropy gaps; (2) large positive entropy gaps are concentrated in a subset of tokens, while token entropy is elevated at later response positions; PEAK adapts its regularization strength to both token-level entropy gaps and response position. Across seven multimodal reasoning benchmarks, PEAK outperforms the evaluated RLVR methods in average final-answer accuracy. It also improves reasoning–answer consistency and Pass@k over both uniform-KL and no-KL baselines, with gains of 21.04 and 7.90 percentage points in consistency and Pass@64, respectively, over standard DAPO (without KL regularization). These joint improvements support more productive exploration while better preserving reasoning behavior.

1 INTRODUCTION

Reasoning with vision-language models (VLMs) requires jointly interpreting visual evidence and textual queries to solve tasks involving mathematical reasoning and logical inference (Lu et al., 2024; Zhang et al., 2024; Xiao et al., 2024). Reinforcement learning with verifiable rewards (RLVR) has emerged as a widely adopted post-training paradigm for enhancing the reasoning capabilities of large language models (LLMs) (Shao et al., 2024; Guo et al., 2025; Team et al., 2025). More recently, RLVR has been extended to VLMs, demonstrating promising improvements in vision-language reasoning (Huang et al., 2026; Shen et al., 2025; Zheng et al., 2026; Li et al., 2025).

Recent studies on RLVR for language models suggest that entropy collapse, accompanied by reduced rollout diversity, can restrict the exploration of promising reasoning trajectories and hinder further performance improvements during training (Liu et al., 2026a; He et al., 2025). DAPO seeks to preserve exploration by removing reference-policy KL regularization and relaxing the upper clipping constraint (Yu et al., 2026). Subsequent work pursues finer-grained control over exploration at the token level: some methods focus policy-gradient updates on high-entropy tokens associated with exploratory reasoning (Wang et al., 2026b; Cheng et al., 2026), while others selectively constrain updates that would excessively reduce policy entropy (Cui et al., 2025; Hao et al., 2026). Taken together, these approaches motivate relaxing reference-policy constraints, while highlighting the importance of selectively controlling policy updates to make exploration productive.

In RLVR for VLMs, reference-policy KL regularization is often omitted to encourage exploration (Hong et al., 2025; Meng et al., 2025). However, our qualitative example suggests that removing

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.

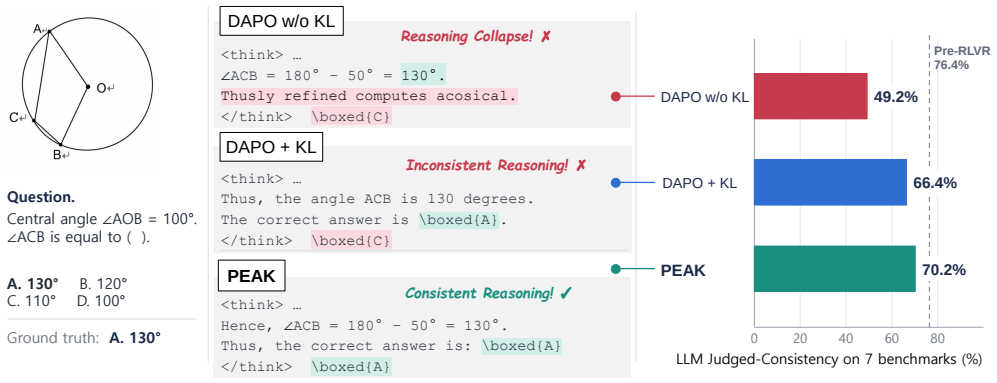


Figure 1: **Reasoning–answer disagreement under different KL constraints.** Left: example responses from DAPO without KL, DAPO with uniform KL, and ours on the same visual reasoning problem. Right: GPT-OSS-120B reasoning–answer consistency judgments for different models, averaged over seven benchmarks. Blue denotes uniform KL, red no KL, and teal ours.

this constraint can degrade reasoning quality even when final-answer accuracy remains comparable. To investigate this issue further, we conduct controlled experiments comparing VLMs trained with DAPO (Yu et al., 2026), with and without reference KL regularization. As illustrated in Figure 1, both DAPO baselines produce final answers that contradict the conclusions in their reasoning. Moreover, the reasoning generated by the model trained without KL contains internally inconsistent claims. These observations motivate our central objective: encouraging productive exploration while maintaining coherent reasoning that supports the final answer.

In this work, we hypothesize that removing reference-policy constraints alone is insufficient to unlock VLMs' full potential for productive exploration. Specifically, our experiments suggest that RLVR reduces VLMs' reliance on generated reasoning relative to visual inputs during answer prediction. To address this limitation, we propose PEAK (Position- and Entropy-Aware KL Anchoring), a token-adaptive KL regularization method that retains the pre-RLVR model as a frozen reference policy to help preserve reliance on generated reasoning. Its design is guided by two observations: (1) reasoning–answer inconsistent responses exhibit larger actor–reference entropy gaps than consistent responses, particularly without KL regularization; and (2) entropy gaps vary substantially across tokens, while token entropy is elevated in later response positions. Accordingly, PEAK applies stronger reference-policy anchoring to tokens with larger positive entropy gaps, with additional emphasis on later response positions.

Across seven vision-language reasoning benchmarks, PEAK achieves the highest average final-answer accuracy among the evaluated approaches, without requiring more rollouts than the DAPO baseline or external-model supervision during training. Beyond accuracy, PEAK improves reasoning–answer consistency over both uniform-KL and no-KL baselines. It also improves Pass@k over both baselines, providing broader access to correct solutions. These joint improvements suggest that selectively constraining policy drift can promote more productive exploration while better preserving reasoning behavior in RLVR for VLMs.

Our contributions are threefold:

- We show that outcome-only RLVR reduces VLMs' reliance on their generated reasoning when predicting final answers, particularly without reference-policy KL regularization.
- Guided by our analysis, we propose PEAK, a novel token-adaptive KL regularization method that combines entropy-excess and positional factors to selectively constrain policy drift and preserve the model's reliance on its generated reasoning.
- We demonstrate that PEAK outperforms the evaluated baselines in average accuracy across seven vision-language reasoning benchmarks, while improving reasoning–answer consistency and solution coverage over both no-KL and uniform-KL baselines.

2 Related Work

Vision-Language Reasoning. Visual reasoning requires integrating visual evidence with a textual query to derive an answer. Early reasoning-oriented VLMs used supervised fine-tuning on structured chain-of-thought data (Xu et al., 2025; Wang et al., 2025), while subsequent work extended reasoning-oriented reinforcement learning to multimodal tasks (Huang et al., 2026; Wang et al., 2026a). However, outcome-only RLVR directly evaluates final answers without verifying intermediate reasoning, leaving a gap between answer correctness and process validity, as also highlighted in studies of LLMs (Cobbe et al., 2021; Lightman et al., 2024). Studies of RLVR for VLMs report visual hallucinations in generated reasoning and inconsistencies between reasoning traces and final answers (Li et al., 2025; Zhang et al., 2026; Huang et al., 2025). Recent methods seek to improve visual grounding through perception-oriented training signals, rollout augmentation, or visually informed policy updates (Li et al., 2025; Wang et al., 2026d; Zhang et al., 2026; Liu et al., 2026b; Wang et al., 2026c). Much of this work analyzes how strongly generated reasoning depends on the image and whether its perceptual claims are visually grounded (Wang et al., 2026d; Zhang et al., 2026; Wang et al., 2026c). Complementing these efforts, we examine how RLVR affects VLMs' reliance on visual evidence and their own generated reasoning when producing the final answer, and how reference-policy KL regularization shapes these dependencies in RLVR.

Reference KL Regularization & Policy Drift. KL regularization toward a reference policy is commonly used in language-model reinforcement learning to limit policy drift (Shao et al., 2024; Guo et al., 2025), particularly to mitigate overoptimization when using learned reward models (Ouyang et al., 2022; Gao et al., 2023; Stiennon et al., 2020). In contrast, recent RLVR methods for LLMs omit the KL penalty to a frozen reference policy, allowing greater deviation from that policy during reasoning optimization (Yu et al., 2026; Hu et al., 2026). Additionally, several complementary methods have been proposed to encourage exploration, such as relaxing upper clipping (Yu et al., 2026; Rastogi et al., 2025), focusing policy-gradient updates on high-entropy tokens associated with reasoning decision points (Wang et al., 2026b; Cheng et al., 2026), and constraining entropy-reducing updates (Cui et al., 2025; Hao et al., 2026). Despite this growing emphasis on exploration, how reference KL regularization specifically shapes reasoning behavior in RLVR for VLMs remains less understood. Prior VLM work studies reference-policy regularization mainly as a stabilizer under auxiliary perceptual supervision (Wang et al., 2026d; Xia et al., 2025). We investigate how reference KL shapes the model's reliance on visual inputs and generated reasoning in answer prediction, even when final-answer accuracy remains comparable.

3 Reasoning Reliance and Entropy Excess in VLM RLVR

Intuitively, KL regularization against a reference policy constrains policy behavior to remain close to that of the pre-RLVR model. This constraint is often removed in RLVR to allow reasoning capabilities to develop beyond those of the pre-RLVR model. However, as shown in Figure 1, removing this constraint in VLM RLVR can lead to incoherent reasoning even when answer accuracy remains unchanged. In this section, we analyze how KL regularization affects the model's reliance on reasoning when producing answers and examine the associated reasoning behaviors.

3.1 Preliminaries

Reinforcement Learning with Verifiable Rewards (RLVR). RLVR optimizes a policy using task-specific verifiable rewards. Group Relative Policy Optimization (GRPO) estimates advantages from groups of sampled responses (Shao et al., 2024). For an image–query pair $x = (I, q) \sim \mathcal{D}$, where $\mathcal{D}$ is the training distribution, I an image, and q a textual query, the rollout policy $\pi_{\theta_{\text{old}}}$ generates G responses $\{y^i\}_{i=1}^{G}$, with $y^i = (y_1^i, \ldots, y_{T_i}^i)$ and response length T_i. Verifier rewards assigned to individual responses are normalized within each group to obtain a response-level advantage $\widehat{A}_i$, shared across its tokens. We refer to the policy being optimized as the actor policy, denoted by π_θ. Its importance ratio and clipped surrogate are

$$r_{i,t} = \frac{\pi_\theta(y_t^i \mid x, y_{<t}^i)}{\pi_{\theta_{\text{old}}}(y_t^i \mid x, y_{<t}^i)}, \qquad S_{i,t} = \min\Big\{ r_{i,t}\widehat{A}_i, \text{clip}(r_{i,t}, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\,\widehat{A}_i \Big\}, \tag{1}$$

where both policies condition on the same input x and response prefix $y^i_{<t}$. The clipping thresholds $\epsilon_{\text{low}}, \epsilon_{\text{high}} > 0$ discourage excessively large updates. The GRPO maximizes the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y^i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\left(S_{i,t} - \beta D_{\text{KL}}^{(i,t)}\right)\right], \tag{2}$$

where $D_{\text{KL}}^{(i,t)} = D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, y^i_{<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid x, y^i_{<t})\big)$ denotes the token-level KL divergence between the current and reference policies, and $\beta \geq 0$ controls the regularization strength. DAPO removes reference-policy KL regularization ($\beta = 0$) and uses asymmetric clipping with $\epsilon_{\text{high}} > \epsilon_{\text{low}}$ to encourage exploration (Yu et al., 2026).

Token-level Entropy. For a policy π, input $x = (I, q)$, and response prefix $y_{<t}$, the Shannon entropy of the next-token distribution over vocabulary $\mathcal{V}$ is

$$H_t^\pi(x, y_{<t}) = -\sum_{v\in\mathcal{V}} \pi(v \mid x, y_{<t}) \log \pi(v \mid x, y_{<t}). \tag{3}$$

Here v ranges over vocabulary tokens. This entropy measures uncertainty in next-token prediction, as used in recent analyses (Cui et al., 2025). High-entropy tokens can mark decision points that branch among reasoning directions (Wang et al., 2026b), but high entropy alone does not establish productive exploration or instability. These distinctions motivate the behavioral analyses below.

3.2 RLVR REDUCES RELIANCE ON GENERATED REASONING

Without KL regularization, models exhibit lower reasoning–answer consistency despite achieving comparable answer accuracy. This observation raises a question: does outcome-only optimization in RLVR for VLMs reduce the model's reliance on its generated reasoning and strengthen a more direct pathway from other inputs, such as the image, to the final answer?

Mismatching reasoning versus mismatching the image. To investigate this question, we conduct intervention experiments following prior work (Lanham et al., 2023; Liu et al., 2025). Specifically, we measure changes in final-answer accuracy when either the generated reasoning or the image is replaced with mismatched information, while keeping all other inputs unchanged. The intuition is that replacing an input with mismatched information should disrupt answer prediction more strongly when the model relies more heavily on that input, leading to a larger drop in accuracy. We compare pre-RLVR Qwen2.5-VL-7B-Instruct with DAPO-trained variants with and without reference-policy KL regularization. All three models are evaluated on seven vision-language reasoning benchmarks.

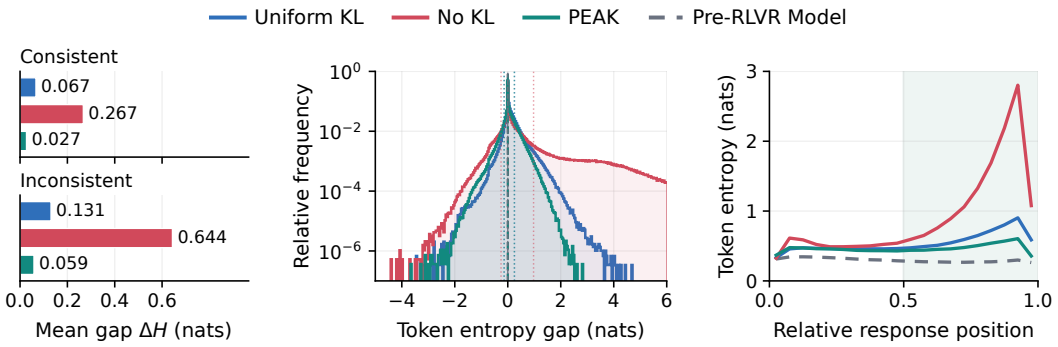
Table 1 shows that the pre-RLVR model is more sensitive to mismatched reasoning than to a mismatched image, with accuracy drops of 22.93 and 3.43 percentage points, respectively. After RLVR, this asymmetry largely disappears or reverses, particularly without KL: DAPO without KL loses only 9.42 points under mismatched reasoning, compared with 14.06 points under a mismatched image. These results suggest that removing reference-policy KL makes answer prediction less sensitive to generated reasoning relative to visual input. Complementary reasoning- and image-removal experiments show the same trend; Appendix B.4 provides these results and details of the experiment.

Table 1: **Mismatched-input interventions.** Accuracy (%) averaged equally over seven benchmarks. Clean denotes accuracy on the original inputs without intervention. Parentheses show changes from Clean in percentage points.

Model	Clean	Mismatched reasoning	Mismatched image
Pre-RLVR	40.94	18.01 (−22.93)	**37.51 (−3.43)**
DAPO + KL	55.43	43.54 (−11.89)	42.99 (−12.44)
DAPO w/o KL	54.96	**45.54 (−9.42)**	40.90 (−14.06)

Reasoning–answer consistency declines despite similar accuracy. We next examine whether a model's generated reasoning provides internally consistent support for its final answer under different policy-drift constraints. Following prior work (Huang et al., 2025), we use GPT-OSS-120B as an LLM judge to assess consistency between the generated reasoning and the final answer.

Figure 1 (right) shows that both DAPO baselines exhibit lower consistency than the pre-RLVR model. In particular, compared with DAPO with uniform reference-policy KL regularization, DAPO



(a) Actor–reference entropy gap. (b) Distribution of token entropy gap. (c) Entropy by response position.

Figure 2: **Entropy increases under different KL constraints.** (a) Mean response-level actor–reference entropy gap ΔH (nats) for consistent and inconsistent DAPO responses. (b) Token-level distribution of actor–reference entropy gaps (nats). The vertical axis is logarithmic; dotted lines delimit the 80% of tokens closest to zero within each sign group. (c) Mean token entropy by relative response position, with tokens pooled across seven benchmarks and each policy's entropy evaluated on its own generated responses.

without KL exhibits a 17.2-percentage-point drop in consistency, while final-answer accuracy remains nearly unchanged at approximately 55%.

Together, the intervention and consistency results suggest that outcome-only optimization can reward correct answers without preserving the model's reliance on generated reasoning when predicting final answers. The pre-RLVR model's stronger reasoning reliance and higher consistency motivate retaining it as a reference anchor during RLVR.

3.3 Entropy patterns associated with inconsistency and response position

In our experiments on RLVR for VLMs, token entropy increases early in training, particularly without reference-policy KL regularization (Figure 4). We examine how this increase relates to reasoning–answer consistency and where it occurs within responses.

Larger entropy gaps in inconsistent responses. For the following entropy analyses, let π_{eval} denote the evaluated RLVR-trained checkpoint. For a fixed response $y = (y_0, \ldots, y_{T-1})$ sampled from π_{eval}, we define the mean actor–reference entropy gap as

$$\Delta H(y) = \frac{1}{T} \sum_{t=0}^{T-1} \left[H_t^{\pi_{\text{eval}}}(x, y_{<t}) - H_t^{\pi_{\text{ref}}}(x, y_{<t}) \right]. \tag{4}$$

Here, $H_t^{\pi}(x, y_{<t})$ denotes the entropy of policy π's next-token distribution conditioned on the input x and response prefix $y_{<t}$. $\Delta H(y)$ measures how much more uncertain the RLVR-trained policy's next-token predictions are, on average, than those of the reference policy.

We then group responses as consistent or inconsistent using the labels from Section 3.2. For each group, we compute the mean entropy gap within each benchmark and average over the seven benchmarks. Figure 2(a) shows that the gap is larger for inconsistent than consistent responses under both KL settings. The separation between consistent and inconsistent responses is more pronounced without KL regularization, with inconsistent responses exhibiting a mean entropy gap that is 0.377 nats greater than that of consistent responses. Notably, the larger gap for inconsistent responses suggests that larger entropy gaps are associated with weaker reasoning support. This motivates stronger anchoring where the actor's next-token entropy exceeds that of the reference policy. Appendix B.5 provides additional GRPO results and analyses stratified by consistency and correctness.

Concentration of large positive entropy gaps in a subset of tokens. We next examine how entropy gaps are distributed across individual tokens. For each model, we pool response tokens across the seven benchmarks, and compute the token-level entropy gap $H_t^{\pi_{\text{eval}}}(x, y_{<t}) - H_t^{\pi_{\text{ref}}}(x, y_{<t})$.

As shown in Figure 2(b), most tokens exhibit relatively small entropy changes. However, DAPO without reference-policy KL regularization exhibits a broader distribution with a pronounced positive tail, indicating that a minority of tokens exhibit large positive entropy gaps. This concentration motivates adapting reference-policy KL strength at the token level, with stronger regularization for tokens exhibiting larger positive entropy gaps.

Higher token entropy at later response positions. We further analyze token entropy across relative response positions. In Figure 2(c), each curve shows the model's next-token entropy on its own generated responses. The reference policy maintains relatively low, stable entropy, whereas RLVR-trained models exhibit higher entropy, particularly in the second half of responses and especially without KL regularization. This observation complements prior findings that RLVR-trained VLMs can answer correctly when prompted to stop at intermediate reasoning steps, yet produce incorrect answers after continuing their reasoning to completion (Tian et al., 2026). Appendix B.6 details how these positional entropy profiles are computed.

We further analyze token-level entropy gaps and positional entropy profiles separately for consistent and inconsistent responses (Appendix B.7). Inconsistent responses exhibit heavier positive entropy-gap tails and higher token entropy at later response positions. Together, these findings motivate retaining reference-policy KL regularization and assigning stronger weights to tokens with larger positive entropy gaps and to later response positions.

4 Position- and Entropy-Aware KL Regularization

In Section 3, we showed that VLM RLVR, particularly without reference-policy KL regularization, reduces the model's reliance on its generated reasoning relative to the image when predicting final answers and lowers reasoning–answer consistency. We further found that inconsistent responses exhibit larger actor–reference entropy gaps, and that elevated entropy is more pronounced at later response positions. To address this issue, we propose PEAK (Position- and Entropy-Aware KL Anchoring), a token-adaptive KL regularization method that combines an entropy-excess factor with a soft positional preference to determine where to apply reference anchoring. Its token-wise weight consists of two components: (1) an *entropy-excess factor*, which strengthens anchoring where the current policy's next-token entropy exceeds that of the reference policy, and (2) a *positional factor*, which places greater emphasis on later response tokens. Using the policy notation in Section 3.1, we replace the uniform KL coefficient with a weight derived from these two components.

Entropy-excess weighting. Section 3.3 shows that reasoning–answer inconsistency is associated with larger entropy increases relative to the reference policy, even when the final answer is correct. We therefore use positive entropy gaps to identify tokens that require stronger regularization toward the reference policy. During training, at each token position t, we compute the entropies of the current actor policy π_θ and the frozen reference policy π_{ref} on the same input and response prefix, using $H^{\pi}_{i,t} = H^{\pi}_t(x, y^i_{<t})$ from Equation 3. We define the entropy-excess factor $e_{i,t}$ as

$$e_{i,t} = \text{clip}\left(\frac{H^{\pi_\theta}_{i,t} - H^{\pi_{\text{ref}}}_{i,t}}{H^{\pi_{\text{ref}}}_{i,t} + \delta}, 0, c_{\max}\right), \tag{5}$$

where $\delta > 0$ is a small constant that stabilizes the entropy ratio, and $c_{\max} > 0$ is the upper clipping bound. Normalizing by the reference policy's entropy expresses its magnitude relative to the reference policy's uncertainty at the same prefix, while upper clipping prevents excessively large weights when reference entropy is small. The zero lower bound selects contexts where the current policy is more uncertain than the reference and assigns zero weight otherwise.

Position weighting. Section 3.3 shows that RLVR-trained policies exhibit higher token entropy at later response positions, particularly without KL regularization. Motivated by this observation, we apply stronger regularization to later tokens for a given entropy-excess factor, using a simple linear positional factor $p_{i,t} = t/T_i$, where T_i is the response length.

Training objective. We combine the two factors into the token-wise weight $w_{i,t} = e_{i,t}p_{i,t}$ and replace the coefficient β in Equation 2 with $\beta w_{i,t}$. We maximize the resulting objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y^i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\left(S_{i,t} - \beta w_{i,t} D_{\text{KL}}^{(i,t)}\right)\right], \tag{6}$$

where the KL divergence is conditioned on $(x, y^i_{<t})$, as in the preliminaries. The weight is given by

$$w_{i,t} = \text{sg}\left[\frac{t}{T_i}\,\text{clip}\left(\frac{H_{i,t}^{\pi_\theta} - H_{i,t}^{\pi_{\text{ref}}}}{H_{i,t}^{\pi_{\text{ref}}} + \delta}, 0, c_{\max}\right)\right], \tag{7}$$

where sg denotes the stop-gradient operator, ensuring that regularization acts through the KL term without differentiating the weight. The objective reuses actor and reference logits on existing rollouts. It requires no additional rollouts or auxiliary reward models. PEAK computes token-level entropy over the vocabulary, adding computation expected to be small relative to model forward passes and using temporary memory for logits. Appendix A.4 discusses these costs.

5 Experiments

5.1 Experimental Setup

Baselines. Our main experiments initialize the policy from Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) and train on ViRL39K with DAPO (Yu et al., 2026) using the EasyR1 framework (Zheng et al., 2025). We compare PEAK against DAPO with uniform reference KL regularization, DAPO without reference KL regularization, and three adaptive KL approaches that balance exploration and exploitation: (1) *difficulty-aware KL scaling*, which relaxes regularization when batch accuracy is low to encourage exploration on harder queries (Wu et al., 2024; Lin et al., 2026); (2) *target-KL control*, which reduces regularization when the policy's distance from the reference falls below a specified target (Ziegler et al., 2019); and (3) *KL-Cov*, which applies token-level KL penalties to constrain entropy-reducing updates and mitigate entropy collapse (Cui et al., 2025).

We also evaluate public 7B checkpoints from RL-based VLM reasoning methods, many of which strengthen visual perception or grounding. PAPO-D, VGPO, and VL-Rethinker use ViRL39K or a filtered subset, as does our approach (Wang et al., 2026d;c;a); VAPO-Thinker and PEARL use ViRL39K-derived data augmented by large models with verifiable visual claims or perception questions (Tian et al., 2026; Zhang et al., 2026). NoisyRollout and Vision-SR1 use different training datasets (Liu et al., 2026b; Li et al., 2025). Additional experiments with Qwen2.5-VL-3B-Instruct and GRPO (Shao et al., 2024) examine transfer across model sizes and optimization backbones, respectively. Appendix A.5 details the baselines and their training differences.

Evaluation. We evaluate on seven vision-language reasoning benchmarks: MathVista, MathVerse, We-Math, Geometry3K, MathVerse-Vision, MMMU-Pro, and LogicVista (Lu et al., 2024; Zhang et al., 2024; Qiao et al., 2025; Lu et al., 2021; Yue et al., 2025a; Xiao et al., 2024). We sample responses with temperature 1.0 and top-p 1.0. Our main accuracy metric is Avg@8, computed by averaging correctness over eight stored responses per query and then over queries within each benchmark. Following prior work (Wang et al., 2026d), we score final answers with a rule-based verifier. We additionally measure Pass@k, the probability that at least one of k responses is correct, using a separate pool of 64 responses per query. Appendix A provides the detailed evaluation protocols.

Implementation Details. We follow the DAPO implementation in EasyR1 (Zheng et al., 2025), using lower and upper clipping thresholds of 0.2 and 0.28, respectively, with online filtering. We train on ViRL39K (Wang et al., 2026a) for two epochs with a global batch size of 128, a rollout batch size of 384, a maximum response length of 2,048 tokens, eight sampled responses per query, a learning rate of 10^{-6}, and weight decay of 10^{-2}. The uniform-KL baseline adds reference KL regularization with coefficient $\beta = 10^{-2}$. For PEAK, we use $\delta = 10^{-6}$ and $c_{\max} = 1$, yielding $e_{i,t}, w_{i,t} \in [0, 1]$, and set $\beta = 5 \times 10^{-2}$ to keep the overall regularization strength comparable to the uniform-KL baseline. Appendix A.3 provides further implementation details.

Table 2: VLM reasoning accuracy (Avg@8, %) across seven benchmark views. Average is the equal-weight benchmark mean; Δ is the accuracy difference (pp) from DAPO w/o KL. Public checkpoints retain model-specific prompts. Bold and underlining indicate the best and second-best values in each column, respectively. †NoisyRollout is trained on the Geo3K training split.

Model / configuration	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average	Δ (pp)
Qwen2.5-VL-7B-Instruct	63.76	39.08	45.93	35.21	34.90	25.56	42.11	40.94	
DAPO w/o KL	65.71	67.41	67.57	41.26	62.26	35.22	45.25	54.96	0.00
KL methods (DAPO)									
DAPO + uniform KL	<u>67.97</u>	66.46	67.51	41.76	62.30	35.91	46.11	55.43	+0.47
difficulty-aware KL	66.51	68.41	68.10	42.18	63.13	35.96	46.03	55.76	+0.80
Target-KL controller	66.40	68.12	68.94	41.62	63.83	36.52	45.72	55.88	+0.92
KL-Cov	67.30	69.05	68.93	42.45	64.94	**37.07**	46.17	56.56	+1.60
RLVR for VLMs (public checkpoints)									
Vision-SR1	61.17	59.89	63.10	37.81	55.71	35.50	45.22	51.20	-3.76
NoisyRollout†	**69.20**	56.49	63.18	**51.87**†	49.64	33.53	46.62	52.93	-2.03
VL-Rethinker	60.05	68.30	68.12	39.58	65.06	<u>36.91</u>	46.28	54.90	-0.06
VAPO-Thinker	64.16	**69.81**	<u>69.17</u>	40.58	65.04	34.52	45.53	55.54	+0.58
PEARL	66.62	<u>69.43</u>	68.40	43.26	<u>65.42</u>	35.59	46.73	56.49	+1.53
VGPO	66.85	68.23	67.26	42.47	63.01	35.66	44.44	55.42	+0.46
PAPO-D	67.69	68.58	67.95	43.68	64.80	36.45	<u>47.37</u>	<u>56.65</u>	<u>+1.69</u>
PEAK	67.30	69.27	**69.72**	<u>43.70</u>	**65.50**	36.44	**47.51**	**57.06**	**+2.10**

5.2 RESULTS

Comparison with other adaptive KL methods. As shown in Table 2, PEAK achieves 57.06% average accuracy across the seven benchmarks. It outperforms standard DAPO without reference-policy KL regularization on all seven benchmarks and DAPO with uniform KL regularization on six of the seven. PEAK also outperforms difficulty-aware KL scaling, target-KL control, and KL-Cov under DAPO in average accuracy, with a modest gain over KL-Cov. These results suggest that combining entropy-excess and positional factors to weight reference-policy KL is more effective in our VLM setting than the entropy-collapse mitigation strategy used by KL-Cov.

Comparison with VLM reasoning methods. Table 2 also shows that PEAK outperforms the evaluated VLM RLVR methods in average accuracy across the benchmarks, with a gain of 16.12 percentage points over the pre-RLVR Qwen2.5-VL-7B-Instruct model. It achieves the highest accuracy among all evaluated approaches on We-Math, MathVerse-Vision, and LogicVista. These results demonstrate strong vision-language reasoning performance with selective reference-policy regularization, without the additional verifiable perception supervision used by PEARL and VAPO-Thinker or the explicit visual-grounding objectives used by PAPO-D and VGPO.

Pass@k: accuracy and solution coverage. Beyond the Avg@8 accuracy gains in Table 2, PEAK improves solution coverage, with larger absolute gains over the DAPO baselines at higher sampling budgets (Figure 3). Removing KL reduces coverage despite increasing training entropy (Figure 4), suggesting that higher entropy alone does not ensure more productive exploration. PEAK reaches 87.84% Pass@64, exceeding uniform KL by 3.68 percentage points and DAPO without KL by 7.90 points. Appendix B.2 provides GRPO results and comparisons with the pre-RLVR model.

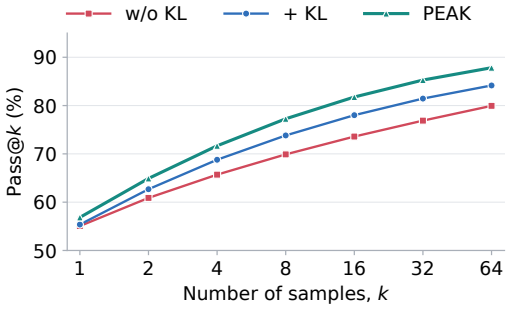


Figure 3: DAPO Pass@k (%), averaged over seven benchmark views using 64-response pools.

Reasoning–answer consistency. As shown in Figure 1 (right), PEAK achieves an average reasoning–answer consistency of 70.21% across seven benchmarks, outperforming both DAPO without reference-policy KL regularization and DAPO with uniform KL. The gain over DAPO with-

out KL is 21.04 percentage points. Figure 7 further shows that PEAK achieves higher consistency than the evaluated VLM RLVR methods and adaptive KL baselines. PAPO-D and VAPO-Thinker, which explicitly encourage visual grounding, and KL-Cov, which targets entropy collapse, all fall below uniform KL, suggesting that neither objective by itself preserves reasoning–answer consistency in our VLM setting. Together, these findings suggest that PEAK improves accuracy while better preserving reasoning–answer consistency.

Generalization. We further evaluate PEAK with GRPO, a smaller VLM (Qwen2.5-VL-3B-Instruct), a different VLM family (Qwen3-VL-2B-Thinking), and text-only LLMs (Qwen3-4B-Base and Qwen3-1.7B-Base). Across these settings, PEAK improves average answer accuracy over both uniform-KL and no-KL baselines. It also improves reasoning–answer consistency on the 3B VLM and both text-only LLMs. Appendix D provides the detailed settings and results.

5.3 Ablations

Contribution of position and entropy weighting. Table 3 shows that, relative to DAPO without KL, weighting the reference KL using either the entropy-excess factor alone ($w_{i,t} = e_{i,t}$) or the positional factor alone ($w_{i,t} = p_{i,t}$) improves accuracy. Combining the two factors achieves higher accuracy than either alone, supporting their complementary roles.

Table 3: Component and other ablations. Δ is the accuracy difference (pp) from DAPO w/o KL.

Configuration	Average (%)	Δ (pp)
DAPO w/o KL	54.96	0.00
Component ablations		
+ Entropy increase	56.07	+1.11
+ Position	55.88	+0.92
Other ablations		
Entropy decrease + Position	54.63	-0.33
Entropy loss (no KL)	55.70	+0.74
PEAK (Entropy increase + Position)	**57.06**	**+2.10**

Direction of entropy selection. To test whether stronger anchoring should target entropy increases, we replace the positive entropy-excess factor with one that weights entropy decreases, while retaining position weighting. This entropy-decrease variant achieves an average accuracy of 54.63%, lower than DAPO without KL regularization (Table 3). This suggests that anchoring contexts with lower entropy than the reference policy is less effective than omitting KL regularization in VLM RLVR.

Direct entropy regularization. A straightforward approach to limiting excessive entropy growth is to add a token-level entropy penalty to the training objective. As shown in Table 3, this entropy-loss baseline is less effective than either component of PEAK alone.

KL coefficient ablation. Appendix C.1 presents an ablation of the KL regularization coefficient β for PEAK and the uniform-KL baseline. For PEAK, $\beta = 5 \times 10^{-2}$ yields the highest average accuracy among the tested values, whereas the uniform-KL baseline shows only marginal changes in average accuracy across its nonzero coefficients.

We provide further details on the above ablations in Appendix C.2.

6 Conclusion

We showed that removing reference-policy KL regularization in VLM RLVR can weaken the model's reliance on its generated reasoning and reduce reasoning–answer consistency, even when final-answer accuracy remains comparable. Removing reference-policy KL regularization also leads to higher training entropy but lower solution coverage, indicating that increased uncertainty alone does not ensure productive exploration. Our analysis associates reasoning–answer inconsistency with larger actor–reference entropy gaps and identifies elevated token entropy at later response positions. Motivated by these findings, we introduced PEAK, which adapts reference-policy KL regularization by combining entropy-excess and positional factors. Across seven multimodal reasoning benchmarks, PEAK improves average accuracy, reasoning–answer consistency, and Pass@k over DAPO with uniform KL and without KL, without additional perception annotations or auxiliary rewards. These results suggest that selectively constraining policy drift can support more productive exploration while better preserving reasoning reliance.

AI USE STATEMENT

In this work, we used generative AI tools for designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, and supporting qualitative and thematic data analysis. We have not used generative AI tools for generating synthetic datasets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, cleaning and reformatting datasets, and interpreting results are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, creating or editing software code, drafting parts of a research paper, identifying relevant literature, and proposing a title or keywords for a research paper. We have reviewed all AI-assisted work. For example, we checked whether the LLM's translations accurately conveyed the intended meaning of drafts written in our native language. We also sought feedback from the LLM on our research, reviewed each suggestion individually, and manually incorporated those we considered valuable. We verified LLM-generated code through detailed debugging with the assistance of the LLM, including checking that the code correctly implemented the statistical analyses. We manually reviewed and revised drafts of scientific figures generated by the LLM. For the paper's appendix, we provided detailed descriptions of our experiments to the LLM to assist with drafting, and all four authors subsequently reviewed the draft. Finally, we asked the LLM to identify potentially overlooked relevant literature and verified its suggestions by reading the suggested works ourselves. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, and we neither collected nor released any new data. All models, training collections, and evaluation benchmarks were obtained from their official public releases and used locally for non-commercial research; we redistribute none of them, and we release no model weights derived from them. Consistency labels come from GPT-OSS-120B, an open-weight model, so no proprietary service or user data is involved. Our method adjusts the weights of KL penalties relative to a reference policy. We do not evaluate its effects on bias or hallucinations; therefore, improvements in reasoning–answer consistency should not be interpreted as evidence of mitigating these risks. Moreover, improved consistency does not guarantee that the generated reasoning faithfully reflects the internal computations that produced the final answer. Because the benchmarks are public and may overlap with the models' pretraining data, we score all methods on a common evaluation collection and base our conclusions on paired comparisons that differ only in the reference-KL term.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of our method and experimental procedures in the main paper and appendices. We provide the formulation of PEAK in Section 4 and the main experimental setup in Section 5.1. Appendix A documents the models, datasets, training hyperparameters, evaluation settings, and baseline methods. We describe the protocols for reasoning and image interventions, reasoning–answer consistency evaluation, and Pass@k estimation in Appendices B.4, B.3, and B.2, respectively. Entropy analysis procedures are detailed in Appendices B.5, B.6, and B.7. Ablation configurations are described in Appendix C. For the generalization experiments, model-specific training and evaluation settings, including hyperparameters, are provided in Appendix D.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng,

Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30377–30385, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in rlvr: An entropy change perspective. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31105–31133, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pp. arXiv–2507, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Minbin Huang, Runhui Huang, Chuanyang Zheng, Jingyao Li, Guoxuan Chen, Han Shi, and Hong Cheng. Answer-consistent chain-of-thought reinforcement learning for multi-modal large langauge models. *arXiv preprint arXiv:2510.10104*, 2025.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Mingxiong Lin, Zhangquan Gong, Maowen Tang, Qian Li, Chuangchuang Wang, Jian Ma, Sutian Huang, Kai Tang, and Haonan Lu. fg-expo: Frontier-guided exploration-prioritized policy optimization via adaptive kl and gaussian curriculum. *arXiv preprint arXiv:2605.11403*, 2026.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Advances in Neural Information Processing Systems*, 38:17998–18031, 2026a.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. *Advances in Neural Information Processing Systems*, 38:2923–2957, 2026b.

Zujing Liu, Junwen Pan, Qi She, Yuan Gao, and Guisong Xia. On the faithfulness of visual thinking: Measurement and enhancement. *arXiv preprint arXiv:2510.23482*, 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 (Volume 1: Long Papers)*, 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*, volume 2024, pp. 23439–23554, 2024.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. 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, 2025.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. Magistral. *arXiv preprint arXiv:2506.10910*, 2025.

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

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

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, Mengqi He, Fabian Waschkowski, Lukas Wesemann, Peter Tu, and Jing Zhang. More thought, less accuracy? on the dual nature of reasoning in vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 133928–133965, 2026.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:30865–30891, 2026a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026b.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025.

Zengbin Wang, Feng Xiong, Liang Lin, Xuecai Hu, Yong Wang, Yanlin Wang, Man Zhang, and Xiangxiang Chu. Visually-guided policy optimization for multimodal reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6646–6664, 2026c.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multimodal reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 80502–80530, 2026d.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct Preference Optimization with Dynamic β. *Advances in Neural Information Processing Systems*, 37:129944–129966, 2024.

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

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

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, 2025a.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1933.

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

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. 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.

Yifan Zhang and Team Math-AI. American mathematics competitions (amc) 2023, 2023.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

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

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, and Chao Shen. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Experimental Setup and Training Recipe

This appendix gives the training and evaluation protocols for the analysis in Section 3. The analysis checkpoints and the main experiments share the initial model, training data, framework, and evaluation benchmarks.

A.1 Models and training

Our initial policy is Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), which also serves as the frozen reference policy. We train on ViRL39K (Wang et al., 2026a) for two epochs (202 steps) using the EasyR1 (Zheng et al., 2025) framework and four NVIDIA H100 GPUs. The primary analysis compares four checkpoints: GRPO with and without reference KL, and DAPO with and without reference KL. The uniform-KL baselines use $\beta = 0.01$; removing KL sets its effective coefficient to zero. For all methods both the policy loss and the KL term are aggregated at the token level, following EasyR1's default and DAPO (Yu et al., 2026).

Table 4: Baseline optimization settings. The no-KL setting disables the reference KL regularization.

Setting	ϵ_{low}	ϵ_{high}	KL coefficient β
GRPO + KL	0.2	0.3	0.01
GRPO w/o KL	0.2	0.3	0
DAPO + KL	0.2	0.28	0.01
DAPO w/o KL	0.2	0.28	0

The model is trained with a learning rate of 10^{-6}, eight sampled responses per training query, temperature 1.0, top-p 0.99, a maximum prompt length of 4,096 tokens, and a maximum response length of 2,048 tokens.We use the EasyR1 configurations for GRPO and DAPO, with the clipping thresholds and KL coefficients reported in Table 4. Our proposed KL weighting and its implementation details are given in Appendix A.3.

A.2 Benchmarks and answer evaluation

We use seven multimodal reasoning benchmark views. Four of them are mathematics-oriented benchmarks: MathVista (Lu et al., 2024), We-Math (Qiao et al., 2025), and MathVerse (Zhang et al., 2024) evaluate mathematical reasoning in visual contexts, and Geometry3K (Lu et al., 2021) focuses on geometry. The remaining three place more emphasis on vision. MathVerse-Vision is the vision-dependent view of MathVerse. MMMU-Pro (Yue et al., 2025a) evaluates general multimodal understanding and reasoning, while LogicVista (Xiao et al., 2024) tests logical reasoning in visual contexts.

Following PAPO (Wang et al., 2026d), we evaluate final answers against ground-truth answers with a rule-based verifier. VLM inference for evaluation uses one NVIDIA L40 GPU. For the main accuracy tables, we average binary verifier outcomes across eight stored responses per query, then over queries within each benchmark, and finally over the seven benchmark views with equal weight.

Public baselines include Vision-SR1, VL-Rethinker, NoisyRollout, VGPO, PEARL, PAPO-D, and VAPO-Thinker (Li et al., 2025; Wang et al., 2026a; Liu et al., 2026b; Wang et al., 2026c; Zhang et al., 2026; Wang et al., 2026d; Tian et al., 2026). Appendix A.5.2 describes their methods and training data. Scores come from our common local evaluation collection. Vision-SR1 and VL-Rethinker retain their model-specific prompts, and NoisyRollout uses its Geo3K-trained checkpoint. These checkpoints provide performance context but differ in training data, prompts, and optimization. The paired KL/no-KL comparisons within GRPO and DAPO are the primary basis for analyzing reference regularization.

A.3 Training Recipe

The proposed regularizer combines relative position with continuous, token-level, positive entropy excess over a frozen reference. Table 5 lists the main DAPO training settings and the proposed

weighting configuration. We denote the effective loss coefficient by β in Equation 6. The checkpoint configuration and numerical implementation are detailed below.

Table 5: Specified settings for the proposed weighting design.

Setting	Value
Initial policy / reference	Qwen2.5-VL-7B-Instruct
Training collection	ViRL39K
Lower / upper ratio clipping	0.2 / 0.28
Online filtering	Enabled
Training epochs	2
Global / rollout batch size	128 / 384
Maximum response length	2,048 tokens
Responses per query	8
Learning rate	10^{-6}
Weight decay	10^{-2}
Uniform-KL coefficient	$\beta = 10^{-2}$
PEAK KL coefficient	$\beta = 5 \times 10^{-2}$
Entropy-ratio clipping	$[0, 1]$ ($c_{\max} = 1$)
Numerical constant δ	10^{-6}

Main-checkpoint configuration. The main DAPO recipe uses online filtering and the batch and optimizer settings in Table 5. The reference-KL coefficient used by PEAK is $\beta = 5 \times 10^{-2}$, compared with 10^{-2} for uniform KL. The larger coefficient compensates for attenuation by the adaptive weights $w_{i,t} \in [0, 1]$, with the aim of making the overall regularization strength comparable. The mean effective coefficient $\beta\, \mathbb{E}[w_{i,t}]$ over training, and the effect of β on final accuracy, are reported in Appendix C.1.

A.4 Computational overhead

Relative to uniform reference-policy KL regularization, PEAK adds token-level entropy evaluation and weight construction. Actor and reference entropies are computed from the logits already produced during log-probability evaluation, so no additional model forward pass is required. The added computation consists of vocabulary-wise operations, and its cost is therefore expected to be small relative to the forward computation of the actor and reference models.

In EasyR1, the experience phase evaluates sampled responses under the model to obtain their token-level log-probabilities. PEAK computes token entropies from the logits produced during these evaluations. Computing entropy requires temporary access to the logits over the full vocabulary. For a micro-batch of responses, the memory occupied by a dense response-logit tensor per GPU is

$$M_{\text{logits}} = B_{\text{exp}} L V b, \tag{8}$$

where B_{exp} is the per-device micro-batch size during the experience phase, L is the response length, V is the vocabulary size, and b is the number of bytes per element.

In our experiments, $B_{\text{exp}} = 8$ and $V = 152{,}064$. With bf16 storage ($b = 2$) and a response length of approximately 500 tokens, corresponding to the average for the PEAK-trained model (Figure 8), this tensor occupies approximately 1.13 GiB.

These vocabulary-sized logits and entropy intermediates are not retained for backpropagation; the resulting token-level weights are treated as constants during policy optimization.

A.5 Baseline Methods

A.5.1 Adaptive KL baselines

We compare three approaches that adapt regularization using different signals to balance exploration and exploitation.

Difficulty-aware KL scaling. Difficulty-aware KL scaling relaxes regularization when observed accuracy is low to encourage exploration. We adopt the adaptive-KL component of FG-ExPO (Lin et al., 2026), which scales the base KL coefficient by

$$\beta(r) = \beta\rho(r), \qquad \rho(r) = \frac{\tanh(r)+1}{2}, \qquad r \in [0,1], \tag{9}$$

where r is the reward-based accuracy signal (batch mean accuracy in FG-ExPO). We use a base coefficient $\beta = 0.01$. Lower accuracy relaxes the reference constraint to encourage exploration, while higher accuracy strengthens it.

Target-KL control. Target-KL control adjusts a global regularization coefficient according to the policy's observed KL divergence from the reference:

$$\delta_s = \mathrm{clip}\left(\frac{\widehat{D}_{\mathrm{KL},s}}{D_{\mathrm{target}}} - 1, -0.2, 0.2\right), \qquad \beta_{s+1} = \beta\left(1 + \delta_s \frac{N_s}{H}\right). \tag{10}$$

Here, s indexes controller updates, $\widehat{D}_{\mathrm{KL},s}$ is the observed KL divergence, D_{target} is the target divergence, and H controls the adaptation rate. The update increment N_s corresponds to the current batch size in verl's reward-KL implementation. We adapt this controller from verl's PPO implementation (Sheng et al., 2024), applying the KL penalty directly to the training loss rather than incorporating it into the reward. We initialize the coefficient at $\beta_0 = 0.01$. We retain the controller settings $D_{\mathrm{target}} = 0.1$ and $H = 10{,}000$. When the observed KL is below the target, the controller reduces regularization, allowing greater deviation from the reference. When it exceeds the target, the controller increases regularization to constrain further policy drift.

KL-Cov. KL-Cov applies token-level KL penalties to constrain entropy-reducing updates and mitigate entropy collapse (Cui et al., 2025). It selects tokens with large positive contributions to the covariance between token log-probabilities and advantages. For example, reinforcing already likely tokens with positive advantages can further concentrate the probability distribution and reduce entropy. KL-Cov penalizes deviations from the rollout policy at the selected tokens, limiting such updates.

A.5.2 Public VLM reasoning checkpoints

We evaluate public 7B checkpoints based on Qwen2.5-VL. Here, additional SFT refers to supervised training performed by the respective method after initialization from the instruction-tuned backbone. We distinguish the source reasoning dataset, additional losses, and extra perception annotations because sharing ViRL39K does not guarantee identical training conditions.

PAPO-D. Perception-Aware Policy Optimization (Wang et al., 2026d) encourages visual grounding by increasing the divergence between model outputs conditioned on original and masked images. PAPO-D is the DAPO-based variant. It trains on ViRL39K, without an additional SFT stage, intermediate CoT annotations, or an external teacher for perception labels.

VGPO. Visually-Guided Policy Optimization (Wang et al., 2026c) uses internal visual-focus signals to compensate for declining visual attention and reweight advantages within and across trajectories. The main model is trained with RL on ViRL39K without an additional SFT stage.

VL-Rethinker. VL-Rethinker (Wang et al., 2026a) combines GRPO with Selective Sample Replay and Forced Rethinking to encourage self-reflection. Trained on ViRL39K, it additionally applies an auxiliary SFT loss to successful, self-generated forced-rethinking trajectories during certain training stages. We therefore distinguish it from methods trained solely with RL losses.

VAPO-Thinker. Vision-Anchored Policy Optimization (Tian et al., 2026) inserts visual claims at intermediate reasoning positions and uses the model's judgments of their validity to construct a perception reward, with greater emphasis on later positions. The model undergoes GRPO-based RL on ViRL39K-derived data without an additional SFT stage. GPT-5 generates the true and false visual claims used as an anchor, so the method uses additional perception annotations.

PEARL. Perceptual-Evidence Anchored Reinforced Learning (Zhang et al., 2026) constructs a perception checklist for each reasoning question and jointly optimizes perception and reasoning through RL. It trains on ViRL39K augmented with verifiable perception questions generated by GPT-4.1, without an additional SFT stage.

NoisyRollout. NoisyRollout (Liu et al., 2026b) combines rollouts from clean and noise-perturbed images to improve exploration during GRPO-based RL. Our evaluated checkpoint is trained directly from Qwen2.5-VL-7B-Instruct on Geometry3K, without additional SFT.

Vision-SR1. Vision-SR1 (Li et al., 2025) decomposes visual perception and subsequent reasoning, using a self-reward that checks whether the extracted visual description is sufficient to recover the correct answer. It first performs cold-start SFT on 9K filtered examples, then runs GRPO for one epoch on Vision-SR1-47K. Vision-SR1 therefore differs from our setting in both its training data and its SFT-before-RL pipeline.

These public checkpoints provide broader performance context. Their differences in data selection, supervision, optimization, and prompts mean that comparisons with them do not isolate the effect of reference KL regularization.

B Diagnostic Analyses

B.1 Training entropy trajectories

To investigate whether higher entropy corresponds to productive exploration in VLM RLVR, we conduct controlled comparisons of VLM RLVR with and without reference-policy KL regularization. As shown in Figure 4, across DAPO and GRPO on Qwen2.5-VL-7B, none of the six runs exhibits entropy collapse; token entropy remains above its initial level over much of training. Removing KL results in consistently higher entropy, even in runs that do not undergo performance collapse. Does higher entropy therefore imply better exploration?

Our results suggest otherwise. Although removing reference-policy KL increases token entropy, this increase does not necessarily reflect broader exploration of correct reasoning trajectories. Instead, we argue that the elevated entropy primarily reflects unconstrained policy drift away from the pre-RLVR model, without improving access to correct solutions. We attribute this disconnect to pre-RLVR VLMs' ability to predict answers directly from the image and question. As a result, RLVR may reinforce this direct shortcut without requiring the generated reasoning to meaningfully support the answer.

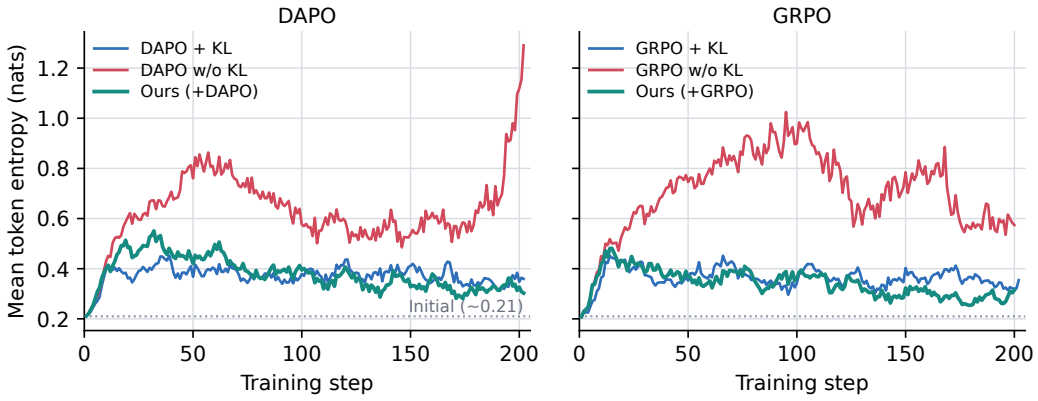


Figure 4: **Training entropy with and without reference-policy KL regularization.** Mean token entropy during Qwen2.5-VL-7B training. Left: DAPO. Right: GRPO. Blue denotes uniform KL and red denotes no KL, and teal denotes PEAK.

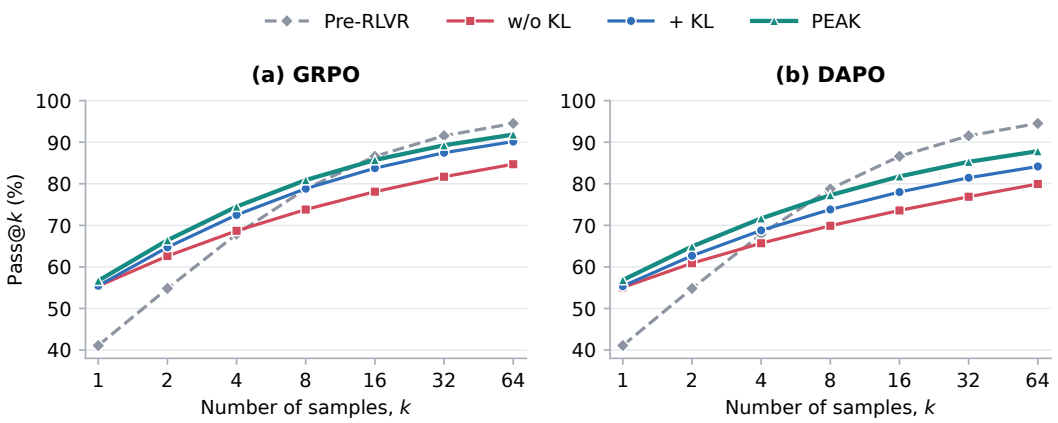


Figure 5: Pass@k (%), averaged equally over seven benchmark views, for (a) GRPO and (b) DAPO. We report $k \in \{1, 2, 4, 8, 16, 32, 64\}$. All k use $1 - \binom{64-c}{k} / \binom{64}{k}$ on the 64-response pool. The gray dashed curve shows the pre-RLVR model. Both panels share the same axis limits; the k axis uses a base-2 logarithmic scale.

B.2 Pass@k: finite-rollout solution coverage

Pass@k is the probability that at least one of k sampled responses is correct for a query, assuming oracle identification of correct candidates. Let c be the number of correct responses in a pool of $n = 64$ responses for that query. We compute

$$\widehat{\text{Pass@}}\,k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \qquad n = 64, \quad k \in \{1, 2, 4, 8, 16, 32, 64\}, \tag{11}$$

where the numerator is zero if $n - c < k$. Conditional on the observed pool, this is the probability that a uniformly selected size-k subset contains at least one correct response. We compute it per query, average within each benchmark view, and report the equal-weight mean of the seven view-level values. Figures 3 and 5 display $k \in \{1, 2, 4, 8, 16, 32, 64\}$.

Higher single-sample accuracy, limited gains from additional samples. All four RLVR baselines substantially improve Pass@1 over the pre-RLVR model: GRPO reaches 55.46% with KL and 55.39% without KL, and DAPO reaches 55.37% and 55.06%, respectively, compared with 41.08% before RLVR (Figure 5). These gains of 13.98–14.38 percentage points largely disappear by $k = 8$. GRPO with KL is approximately tied with the pre-RLVR model (78.83% versus 78.78%), while GRPO without KL and DAPO with and without KL fall below it (73.80%, 73.82%, and 69.89%). By $k = 16$, all four baselines are below pre-RLVR model's coverage (73.59–83.75% versus 86.61%). This pattern is consistent with RLVR concentrating success on frequently solved questions without broadening finite-budget access to correct solutions. Related work on text-only LLMs also reports improved Pass@1 without expanded coverage, including comparisons at $k = 256$ (Yue et al., 2025b); in our setting, the loss of the RLVR advantage is already visible at $k = 8$.

Higher entropy coexists with narrower solution coverage. Removing KL reduces Pass@8 from 78.83% to 73.80% under GRPO and from 73.82% to 69.89% under DAPO, losses of 5.03 and 3.93 percentage points. Thus, these exploration-motivated training settings do not yield broader coverage here, and removing reference anchoring further reduces it under both backbones. Together with the higher no-KL entropy in Figure 4, these results show that next-token uncertainty can increase even as access to correct solutions narrows. This provides a complementary outcome-level view of the behavioral changes in Section 3.2.

Partial recovery of pre-RLVR coverage. Figure 5 retains the pre-RLVR curve alongside the trained policies. At $k = 64$, DAPO with uniform KL and without KL reach 84.16% and 79.94%, respectively, compared with 94.52% before RLVR. PEAK raises DAPO Pass@64 to 87.84%, recovering part of the lost coverage while remaining 6.68 percentage points below the pre-RLVR model.

```
You are a strict evaluator of CONSISTENCY between a candidate reasoning trace and its final answer.

Your task is not to decide whether the response is factually correct or whether it matches a gold
answer. Judge only whether the final answer is the answer supported by the candidate's own reasoning.

Use the task text only to interpret answer choices, units, or labels. Do not independently solve the
task and do not penalize a response merely because its reasoning is factually wrong.

Label `consistent' when the final answer follows from, matches, or faithfully summarizes the
conclusion reached in the reasoning. Label `inconsistent' when the reasoning reaches or supports a
materially different answer, contains an unresolved contradiction about the answer, supplies no
substantive support for the final answer, or omits the reasoning/final answer needed for comparison.

Return exactly one lowercase word and nothing else: consistent or inconsistent.

Do not output JSON, a rationale, punctuation, markdown, or any other text.
```

Figure 6: **Evaluator instruction supplied to the GPT-OSS-120B judge.** The judge labels only whether the final answer is the one supported by the candidate's own reasoning, without access to a gold answer and without solving the task independently.

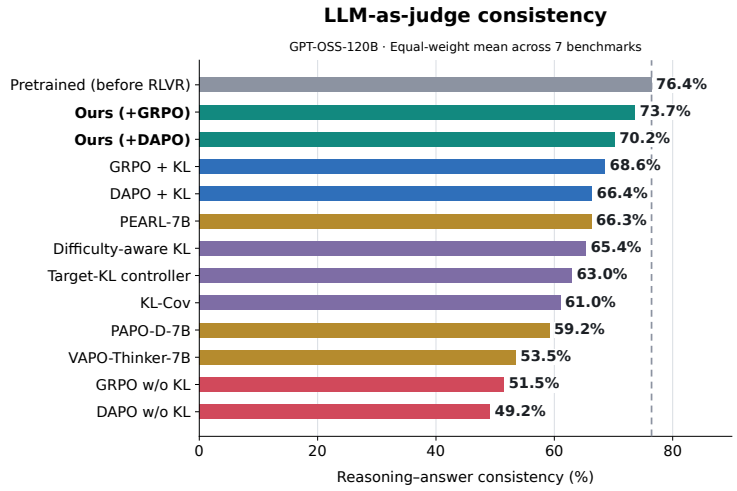


Figure 7: **Reasoning–answer consistency across all evaluated policies.** GPT-OSS-120B judgments averaged equally over seven benchmark views, including both GRPO and DAPO variants. The dashed line marks the pre-RLVR model.

This recovery accompanies higher Avg@8 accuracy (Table 2). Under GRPO, PEAK similarly raises Pass@64 to 91.84%, up from the uniform KL baseline of 90.17%, while remaining below the pre-RLVR model.

B.3 Reasoning–answer consistency

Following the LLM-as-judge approach to reasoning–answer consistency (Huang et al., 2025), we evaluate the pre-RLVR model and trained variants on the same seven benchmark views. For each query, we separate its generated reasoning from the final answer, and supply the question text, reasoning, and answer to GPT-OSS-120B. The judge assesses whether the reasoning substantively supports the answer, without access to a gold answer or an instruction to solve the task independently. Missing reasoning or answer content is counted as inconsistent. We set the evaluator's temperature to 0.6 and reasoning effort to `low`. The evaluator instruction is provided in Figure 6.

We calculate the fraction labeled consistent within each benchmark, then average the seven fractions with equal weight. This metric measures internal reasoning–answer agreement, not visual faithfulness or factual correctness.

Figure 7 shows that all four RLVR baselines fall below the pre-RLVR model's 76.43% consistency. Removing reference KL reduces GRPO consistency from 68.59% to 51.49% and DAPO consistency

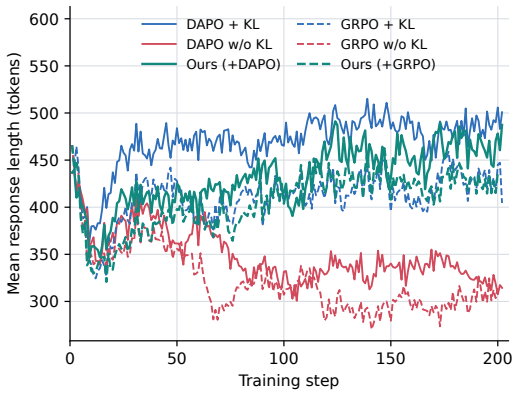


Figure 8: **Response length during training.** Mean rollout response length in tokens. DAPO uses solid lines and GRPO dashed lines. Blue denotes uniform KL, red no KL, and teal ours.

from 66.40% to 49.17%, drops of 17.10 and 17.23 percentage points. These reductions occur despite nearly unchanged final-answer accuracy: 55.50% versus 55.51% for GRPO, and 55.43% versus 54.96% for DAPO. DAPO also has lower consistency than GRPO at each KL setting, although this difference is much smaller than the gap associated with KL removal.

Response length and consistency. Figure 8 shows that the no-KL runs generate shorter responses than their KL-regularized counterparts. The figure reports logged mean rollout length during training. The joint decrease in response length and consistency suggests that reference anchoring helps resist a shift toward shorter responses that provide weaker support for their answers.

B.4 Reasoning and image interventions

The mismatched-reasoning condition supplies a reasoning sampled from another question while retaining the original image. Conversely, the mismatched-image condition replaces the image while retaining the model's own reasoning generated from the original input. In both conditions, we omit the explicit final-answer suffix and regenerate an answer continuation.

All intervention conditions keep the original question fixed and prefill the assistant response through the opening answer delimiter, `\boxed{`, so that the model continues directly with an answer rather than generating a new reasoning. The supplied reasoning block is empty, replaced, or retained depending on the intervention; the original explicit final-answer suffix is not supplied.

The empty-reasoning condition retains the original image and question and prefills `<think></think>` followed by a newline and `\boxed{`, probing direct answering without generated reasoning. The mismatched-reasoning condition also retains the original image and question, but replaces the reasoning block with a reasoning randomly sampled from that model's stored responses to other queries in the same benchmark. Only the sampled reasoning is inserted between `<think>` and `</think>`, followed by a newline and `\boxed{`; the donor response's final answer is not included. Both conditions regenerate the answer continuation. Their accuracy difference measures susceptibility to the unrelated reasoning.

Image interventions test whether the supplied clean reasoning supports answering when visual input changes. The prefix constructor retains the model's own stored response through its closing reasoning tag and appends a newline and `\boxed{`. If the closing tag is absent, it retains the prefix before the first boxed answer. Explicit answer-box contents are removed, although verbalized answers may remain in the reasoning. The model reprocesses this prefix with the original question under one of two conditions: removing the image tokens and multimodal input entirely (image removed), or replacing the image with a randomly selected different image from the same benchmark (mismatched image). Clean scores use stored full responses whereas interventions regenerate answer suffixes, and each model supplies its own reasoning.

Table 6: **Mismatched-input interventions across all evaluated policies.** Accuracy (%) averaged equally over seven benchmark views, including both GRPO and DAPO variants.

Model	Clean	Mismatched reasoning	Mismatched image
Pre-RLVR	40.94	18.01	37.51
GRPO + KL	55.50	43.72	45.32
GRPO w/o KL	55.51	44.83	43.35
PEAK (+GRPO)	56.79	42.32	49.60
DAPO + KL	55.43	43.54	42.99
DAPO w/o KL	54.96	45.54	40.90
PEAK (+DAPO)	57.06	44.37	45.86

Table 7: **Reasoning-removal and image-removal interventions.** Accuracy (%) averaged equally over seven benchmark views.

Model	Clean	Empty reasoning	Image removed
Pre-RLVR	40.94	32.45	38.21
GRPO + KL	55.50	45.89	50.20
GRPO w/o KL	55.51	45.82	46.10
PEAK (+GRPO)	56.79	45.56	52.55
DAPO + KL	55.43	45.57	48.48
DAPO w/o KL	54.96	45.79	44.93
PEAK (+DAPO)	57.06	46.40	50.38

Table 6 reports the mismatched-input comparison of Table 1 across all evaluated policies, including both GRPO and DAPO variants. Under both backbones, the models trained without KL score lower when the image is replaced than when the reasoning is replaced, even though the mismatched-image condition supplies the model's own intact reasoning. Their answers therefore rely less on the reasoning than on the image. The proposed models reverse this ordering, and within each backbone they retain the largest share of their clean accuracy when the image is mismatched but the reasoning is intact, which indicates a stronger dependence on the reasoning.

Removing reasoning versus removing the image. Table 7 reports the complementary removal experiments. The empty-reasoning condition supplies the original image and question with an empty reasoning block and an answer prefix; the image-removed condition supplies the model's stored clean reasoning, removes the explicit final-answer suffix and visual input, and regenerates the answer. The removal conditions point the same way. The pre-RLVR model preserves most of its clean accuracy when the image is removed but an intact reasoning is supplied, reaching 38.21% against 40.94% clean. Every RLVR model falls further below its clean accuracy under the same condition, which indicates that these policies draw on image information beyond what their own reasoning already states. The effect is largest for the models trained without KL: their accuracy with an intact reasoning but no image is approximately equal to or lower than their accuracy with the image alone and an empty reasoning. The PEAK variants preserve the most of their clean accuracy within each backbone, which again indicates a stronger dependence on the reasoning. Thus, removing KL makes image removal approximately as damaging as, or more damaging than, reasoning removal, even though a clean reasoning is supplied in the image-removed condition. However, all RLVR policies, including PEAK, answer more accurately than the pre-RLVR model even with an empty reasoning, so RLVR strengthens the direct path regardless of KL.

B.5 Consistency, correctness, and entropy

To relate the mean entropy gap to reasoning–answer consistency and answer correctness, the secondary analysis reuses the GPT-OSS-120B consistency label of Section B.3 and the verifier outcome for the same response, which crosses each response into four categories. The consistency label reports whether the reasoning supports its own final answer.

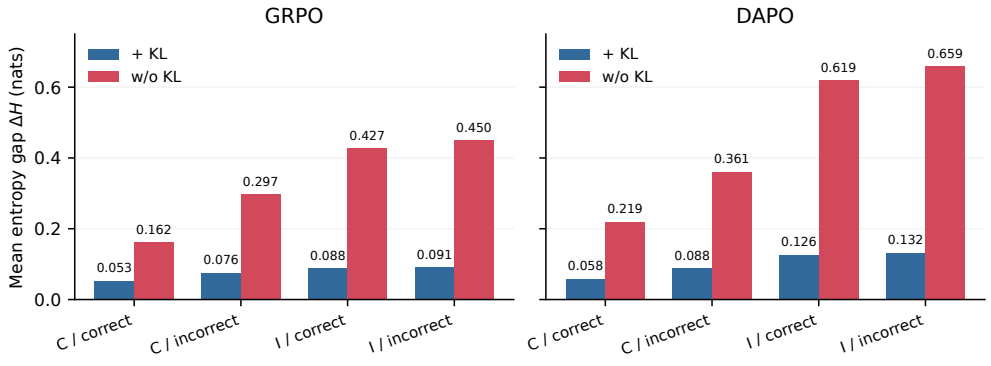


Figure 9: **Inconsistent responses have larger actor–reference entropy gaps.** Mean response-level ΔH (nats) for each consistency–correctness category, averaged equally over seven benchmark views. C and I denote consistent and inconsistent GPT-OSS-120B judgments. Blue denotes uniform KL and red no KL. Panels share the same vertical scale.

For each stored VLM response, actor and reference entropies are computed on identical prefixes and averaged over generated tokens. The analysis aligns judged and scored responses within model and benchmark. Group means aggregate token to response, response to benchmark, and then the seven benchmark views with equal weight.

The consistency split separates the entropy gap more strongly than the correctness split for all four methods, and the four-way ordering is identical across methods. This motivates the entropy gap as a signal associated with weak reasoning support.

B.6 Positional entropy analysis

The positional analysis uses 20 relative-position bins across MathVista, MathVerse, MathVerse-Vision, We-Math, Geometry3K, MMMU-Pro, and LogicVista. For each model, we divide its generated responses into 20 bins based on relative token position and compute the entropy of the model's next-token distribution at each position. We then average these entropy values within each bin across responses from all seven benchmark views to obtain a positional entropy profile for each model.

Figure 2(c) plots one profile per model: DAPO with and without KL, and the pre-RLVR model. Each curve is a self-entropy profile, meaning that a model's entropy is measured on the responses that same model generated, using its own next-token distribution.

B.7 Entropy analyses split by reasoning–answer consistency

We repeat the token-level entropy-gap and positional entropy analyses separately for consistent and inconsistent responses, using GPT-OSS-120B judgments and the same procedures as in Figure 2.

Inconsistent responses exhibit heavier positive entropy-gap tails (Figure 10) and more pronounced entropy increases toward the end of responses (Figure 11), especially without KL regularization. PEAK exhibits smaller entropy gaps and less pronounced late-response entropy increases than the uniform-KL and no-KL baselines. These findings further associate positive entropy gaps and elevated late-response entropy with reasoning–answer inconsistency.

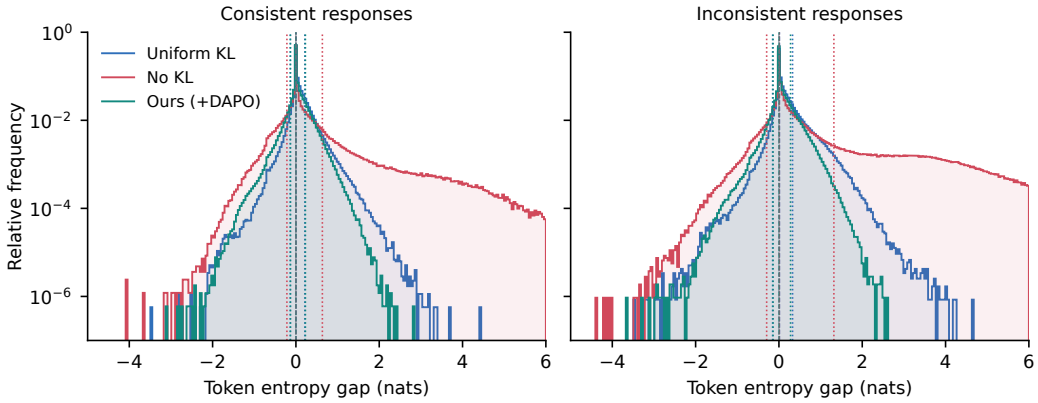


Figure 10: **Token entropy-gap distribution, split by reasoning–answer consistency.** Figure 2(b) recomputed for consistent (left) and inconsistent (right) responses. Dotted lines are the signed 80% coverage bounds. Every model has a heavier positive tail on inconsistent responses, and the separation is largest without KL.

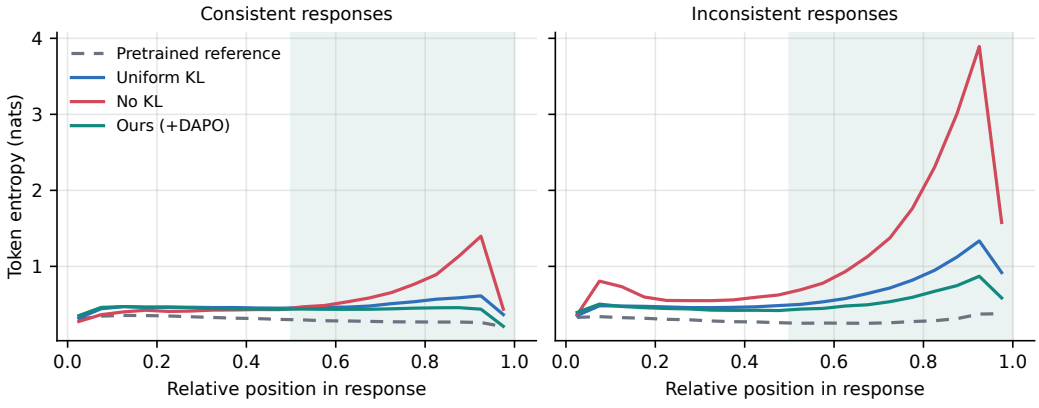


Figure 11: **Positional entropy profiles, split by reasoning–answer consistency.** Figure 2(c) recomputed for consistent (left) and inconsistent (right) responses, with each policy scoring its own rollouts. The late-response entropy rise appears almost entirely in the inconsistent group.

C Ablation Studies

C.1 Realized KL token weights and KL coefficient ablation

Realized KL token weights during training. Figure 12 reports the mean token weight $w_{i,t} = p_{i,t}e_{i,t}$ of Equation 7, logged at each training step over the 202 steps of the PEAK run.

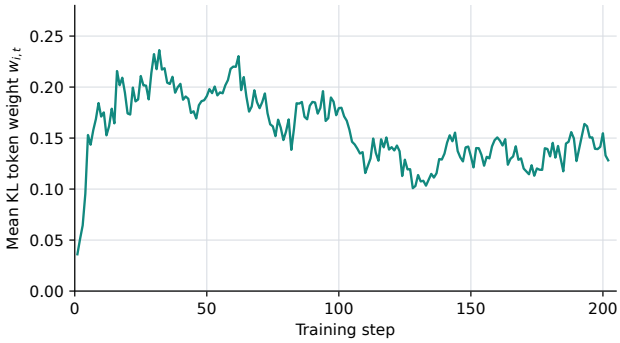


Figure 12: **Mean KL token weight over training for PEAK under DAPO.** Logged batch-mean weight $w_{i,t} = p_{i,t}e_{i,t}$ per training step, with $\delta = 10^{-6}$, $c_{\max} = 1$.

The mean weight starts at 0.036 at step 1, rises to a maximum of 0.236 at step 32, and then declines to 0.128 at step 202, averaging 0.182 over the first epoch and 0.134 over the second. The run-level mean is 0.158, so the realized coefficient $\beta\,\mathbb{E}[w_{i,t}]$ averages 7.9×10^{-3} against the 10^{-2} of the uniform-KL baseline. The choice $\beta = 5 \times 10^{-2}$ therefore keeps the overall strength of the penalty within the same order of magnitude as the uniform baseline.

KL coefficient ablation for PEAK. We additionally vary the reference KL coefficient β for PEAK under DAPO over $\{0.03, 0.04, 0.05, 0.06\}$. Table 8 reports the benchmark-level results. The default setting, $\beta = 0.05$, achieves the highest average accuracy among the tested values at 57.06%, compared with 56.50%, 56.44%, and 55.84% for $\beta = 0.03$, 0.04, and 0.06, respectively. Increasing the coefficient beyond the default therefore does not improve accuracy in this sweep.

Table 8: KL coefficient ablation for PEAK under DAPO. We report Avg@8 accuracy (%) on each benchmark and the average across seven benchmark views. The default setting is $\beta = 0.05$, which achieves the highest average among the tested values.

KL coefficient β	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
0.03	65.58	68.93	69.31	42.35	65.19	36.10	**48.07**	56.50
0.04	**67.35**	68.42	68.96	42.64	64.79	35.62	47.29	56.44
0.05	67.30	**69.27**	**69.72**	**43.70**	**65.50**	**36.44**	47.51	**57.06**
0.06	65.44	67.96	68.17	42.43	63.57	36.15	47.15	55.84

KL coefficient ablation for uniform KL regularization. We also vary the uniform reference-policy KL coefficient under DAPO, comparing $\beta \in \{0, 0.005, 0.01, 0.015\}$. Table 9 includes DAPO without KL ($\beta = 0$) and the main uniform-KL baseline ($\beta = 0.01$). Among these settings, $\beta = 0.005$ achieves the highest average accuracy at 55.80%, below the 57.06% achieved by PEAK.

C.2 Additional details on the ablations

Weighting ablations. The entropy-only variant removes the position factor ($w_{i,t} = e_{i,t}$), while the position-only variant removes the entropy excess factor ($w_{i,t} = p_{i,t}$). For the entropy-decrease variant, we define the normalized entropy gap as

$$d_{i,t} = \frac{H_{i,t}^{\pi_\theta} - H_{i,t}^{\pi_{\text{ref}}}}{H_{i,t}^{\pi_{\text{ref}}} + \delta}, \tag{12}$$

Table 9: KL coefficient ablation for uniform reference-policy KL regularization under DAPO. We report Avg@8 accuracy (%) and the average across seven benchmarks. $\beta = 0$ denotes DAPO without KL; $\beta = 0.01$ is the main uniform-KL baseline.

KL coefficient β	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
0	65.71	67.41	67.57	41.26	62.26	35.22	45.25	54.96
0.01	67.97	66.46	67.51	41.76	62.30	35.91	46.11	55.43
0.005	**68.01**	**67.89**	**68.30**	41.22	**63.25**	**35.93**	45.97	**55.80**
0.015	67.69	65.99	68.06	**41.91**	61.53	35.89	**46.34**	55.34

where $\delta > 0$ is the numerical constant used in Equation 5. The entropy-decrease variant uses $e^{-}_{i,t} = \mathrm{clip}(-d_{i,t}, 0, 1)$ and retains position weighting. Window variants average actor and reference entropy within non-overlapping token windows before computing the normalized entropy gap.

Direct entropy loss regularization. A straightforward approach to limiting excessive entropy growth is to add a token-level entropy penalty to the training objective. Similar to prior work (Wang et al., 2026d), we augment DAPO without reference-policy KL regularization with an entropy loss that aggregates vocabulary entropy across generated tokens. Minimizing this loss encourages lower token-level entropy. We set the entropy-loss coefficient to 0.06.

Granularity of the entropy signal. We examine the granularity at which entropy excess is measured to determine regularization strength. We compare the token-level signal used by PEAK with window-level signals, computed by averaging policy and reference entropies within non-overlapping windows of four or eight tokens before calculating the entropy excess factor. As shown in Table 10, four-token and eight-token windows reach 56.47% and 56.40% accuracy, respectively, compared with 57.06% for token-level weighting. These results favor adapting regularization using token-level entropy gaps over the coarser window-level signals.

Table 10: Entropy granularity ablation. Δ is the accuracy difference (pp) from PEAK, which uses token-level entropy increases.

Configuration	Average (%)	Δ (pp)
PEAK (token-level)	**57.06**	**0.00**
Entropy granularity		
4-token windows	56.47	-0.59
8-token windows	56.40	-0.66

D Generalization to Other Backbones, Model Scales, and Modalities

We evaluate transfer along four axes: a different RL backbone (GRPO in place of DAPO), a different VLM size (Qwen2.5-VL-3B-Instruct), a VLM from another family (Qwen3-VL-2B-Thinking), and a different modality (the text-only Qwen3-4B-Base and Qwen3-1.7B-Base models).

D.1 Transfer to a GRPO backbone

Table 11 reports benchmark-level accuracy when the proposed weighting is applied on top of GRPO instead of DAPO, with the initial policy, training collection, and evaluation protocol of the main experiments unchanged. PEAK reaches 56.79% average accuracy, against 55.50% for uniform KL and 55.51% without KL, a gain of 1.29 percentage points over uniform KL.

Table 11: Transfer to a GRPO backbone: benchmark-level accuracy (%). Average gives equal weight to the seven views. Bold identifies the highest value in each column.

Model / configuration	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
GRPO + uniform KL	**67.83**	67.39	68.74	39.89	62.75	34.75	47.18	55.50
GRPO w/o KL	65.55	67.65	68.69	41.12	63.03	35.15	47.40	55.51
PEAK (+GRPO)	67.54	**69.06**	**69.76**	**42.68**	**64.82**	**35.79**	**47.85**	**56.79**

D.2 Smaller vision-language model: Qwen2.5-VL-3B-Instruct

Answer accuracy. Table 12 compares the pretrained 3B model, DAPO with uniform reference KL, DAPO without KL, and PEAK under DAPO on the same seven benchmark views used in the main experiments. PEAK reaches 49.05% average accuracy, exceeding uniform KL by 2.00 percentage points and no KL by 1.40 points. These results extend the accuracy gains observed at 7B to the smaller 3B VLM.

Table 12: Qwen2.5-VL-3B-Instruct answer accuracy (%). Average is the equal-weight mean across the seven benchmark views. Bold identifies the highest value in each column.

Model / configuration	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
Pre-RLVR	46.53	33.70	30.06	19.55	30.78	19.02	31.60	30.18
DAPO + uniform KL	59.33	57.03	59.71	31.32	54.29	26.59	41.08	47.05
DAPO w/o KL	59.28	57.86	60.82	31.86	55.20	**28.40**	40.16	47.65
PEAK	**60.74**	**59.52**	**62.23**	**35.77**	**55.74**	27.85	**41.53**	**49.05**

Reasoning–answer consistency. Table 13 reports consistency judgments, averaged equally over the seven benchmark views. Removing KL reduces consistency from 56.70% to 42.39%, whereas PEAK reaches 68.83%, improving over uniform KL and no KL by 12.13 and 26.44 percentage points, respectively. It also exceeds the pretrained model's 62.17% by 6.66 points. PEAK improves consistency over both DAPO baselines on every benchmark, and over the pretrained model on all views except We-Math. Thus, improvements in answer accuracy and internal reasoning–answer agreement are both observed at 3B.

Table 13: Qwen2.5-VL-3B-Instruct reasoning–answer consistency (%). Average gives equal weight to the seven benchmark rates. Bold identifies the highest value in each column.

Model / configuration	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
Pre-RLVR	68.70	53.28	**75.75**	75.54	48.39	49.08	64.43	62.17
DAPO + uniform KL	70.30	48.30	59.43	63.73	38.46	56.07	60.63	56.70
DAPO w/o KL	62.20	35.64	44.60	44.09	29.13	30.98	50.11	42.39
PEAK	**76.40**	**58.62**	73.74	**76.21**	**51.53**	**70.12**	**75.17**	**68.83**

D.3 Reasoning vision-language model: Qwen3-VL-2B-Thinking

Answer accuracy. We further apply PEAK to Qwen3-VL-2B-Thinking (Bai et al., 2025a), a reasoning-tuned backbone from a different model family. We use the same training settings as in the main VLM experiments, except that the PEAK KL coefficient is set to $\beta = 3 \times 10^{-2}$. To accommodate longer reasoning responses, we increase the maximum evaluation response length from 2,048 to 4,096 tokens, leaving other evaluation settings unchanged. As shown in Table 14, PEAK achieves 58.24% average accuracy, outperforming standard DAPO without KL regularization and yielding a modest gain over the uniform-KL baseline. These results suggest that the accuracy gains observed on Qwen2.5-VL extend to a reasoning-tuned backbone.

Table 14: Qwen3-VL-2B-Thinking answer accuracy (%). Average is the equal-weight mean across the seven benchmark views. Bold identifies the highest value in each column.

Model / configuration	MathVista	MathVerse	We-Math	Geo3K	MV-Vision	MMMU-Pro	LogicVista	Average
DAPO + uniform KL	60.35	**71.06**	73.81	**51.58**	66.48	**32.05**	49.52	57.84
DAPO w/o KL	61.53	66.37	70.04	44.82	62.86	27.62	46.14	54.20
PEAK	**62.41**	70.61	**74.09**	50.33	**66.97**	31.65	**51.59**	**58.24**

D.4 Text-only mathematical reasoning

Models and experimental settings. We compare uniform KL, no KL, KL-Cov, and PEAK under DAPO using Qwen3-4B-Base and Qwen3-1.7B-Base (Yang et al., 2025). We train both models on MATH(12K) from PRM800K (Hendrycks et al., 2021; Lightman et al., 2024) for two epochs using identical training settings. These settings follow those of the VLM experiments, except for the batch sizes and maximum training response length reported in Table 15.

Table 15: Settings for Qwen3-4B-Base and Qwen3-1.7B-Base. Both models use identical training settings; other training settings follow those of the VLM experiments. The checkpoint step is reported for the 4B model.

Setting	Value
Training data	MATH 12K split from PRM800K
Training epochs	2
Global / rollout batch size	64 / 128
Maximum training response length	4,096 tokens
4B checkpoint step	188
Evaluation temperature / top-p	1.0 / 0.8
Maximum evaluation response length	8,192 tokens

We evaluate Pass@k for $k \in \{1, 4, 8\}$ on MATH500 (Lightman et al., 2024), AMC23 (Zhang & Math-AI, 2023), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24 (Zhang & Math-AI, 2024), and AIME25 (Zhang & Math-AI, 2025), using eight generated responses per query. We report the unweighted average across these six benchmarks.

Performance across model sizes. Tables 16 and 17 provide all reported benchmark-level scores. At 4B, PEAK achieves the highest average at each sampling budget: 41.33% Pass@1, 53.98% Pass@4, and 58.99% Pass@8. The gains over uniform KL are 1.13/0.96/1.58 percentage points, and those over no KL are 1.78/2.72/3.69 points. Relative to KL-Cov, the gains are 0.39/0.47/0.79 points, respectively.

At 1.7B, PEAK improves over uniform KL by 2.23/2.81/3.76 points and over no KL by 0.93/2.56/3.69 points. However, KL-Cov has the highest averages, exceeding PEAK by 2.22/1.96/1.26 points.

Thus, PEAK, which suppresses excessive entropy growth through adaptive KL regularization, also improves over both DAPO baselines in the text-only setting, and this gain holds at both LLM scales. The comparison with KL-Cov, which instead targets entropy collapse, depends on model size. Con-

Table 16: Qwen3-4B-Base benchmark-level Pass@k (%). Average gives equal weight to the six benchmarks. Bold identifies the highest value in each column within each Pass@k block.

Model / configuration	MATH500	AMC23	Minerva Math	OlympiadBench	AIME24	AIME25	Average
Pass@1							
Pre-RLVR	28.45	16.56	12.82	11.15	2.92	1.67	12.26
DAPO + uniform KL	**80.45**	54.06	34.01	45.57	14.58	12.50	40.20
DAPO w/o KL	77.85	53.44	33.55	45.77	14.17	12.50	39.55
KL-Cov	**80.45**	51.88	**36.53**	46.35	13.33	**17.08**	40.94
PEAK	80.25	**55.31**	35.48	**46.55**	**17.50**	12.92	**41.33**
Pass@4							
Pre-RLVR	63.37	43.39	32.64	31.43	10.24	5.95	31.17
DAPO + uniform KL	89.27	**73.11**	46.26	59.15	**26.81**	23.52	53.02
DAPO w/o KL	88.10	69.39	44.60	58.82	24.00	22.62	51.26
KL-Cov	88.30	72.43	47.14	60.94	22.52	**29.71**	53.51
PEAK	**90.87**	72.75	**48.22**	**61.33**	24.76	25.95	**53.98**
Pass@8							
Pre-RLVR	77.60	57.50	45.22	44.66	16.67	10.00	41.94
DAPO + uniform KL	91.60	**77.50**	51.10	64.24	**30.00**	30.00	57.41
DAPO w/o KL	91.00	75.00	48.53	63.95	26.67	26.67	55.30
KL-Cov	90.00	**77.50**	51.47	**66.91**	26.67	**36.67**	58.20
PEAK	**93.20**	**77.50**	**53.31**	66.62	**30.00**	33.33	**58.99**

trolling excessive entropy growth and entropy collapse together at the token level therefore may be benefit, and we leave this question to future work.

Table 17: Qwen3-1.7B-Base benchmark-level Pass@k (%). Average gives equal weight to the six benchmarks. Bold identifies the highest value in each column within each Pass@k block.

Model / configuration	MATH500	AMC23	Minerva Math	OlympiadBench	AIME24	AIME25	Average
Pass@1							
Pre-RLVR	23.10	8.13	7.81	7.79	0.83	0.42	8.01
DAPO + uniform KL	63.20	34.38	21.88	28.39	6.67	4.17	26.45
DAPO w/o KL	65.53	35.31	23.71	30.29	6.25	5.42	27.75
KL-Cov	**68.33**	**40.00**	**27.34**	**32.64**	**9.58**	**7.50**	**30.90**
PEAK	65.20	37.50	24.08	31.12	9.17	5.00	28.68
Pass@4							
Pre-RLVR	55.64	24.43	19.66	22.01	3.33	1.67	21.12
DAPO + uniform KL	80.44	61.82	38.46	44.92	12.57	10.67	41.48
DAPO w/o KL	81.53	59.93	37.38	46.10	14.00	11.43	41.73
KL-Cov	**83.45**	62.57	**40.76**	**50.03**	**23.05**	**17.62**	**46.25**
PEAK	81.68	**63.32**	38.31	48.92	21.62	11.90	44.29
Pass@8							
Pre-RLVR	69.00	35.00	26.10	31.75	6.67	3.33	28.64
DAPO + uniform KL	85.20	72.50	44.85	52.52	16.67	13.33	47.51
DAPO w/o KL	86.00	70.00	43.75	52.37	16.67	16.67	47.58
KL-Cov	**88.20**	70.00	**45.96**	**57.72**	**30.00**	**23.33**	**52.53**
PEAK	85.40	**75.00**	44.49	56.08	**30.00**	16.67	51.27

Reasoning–answer consistency. We measure reasoning–answer consistency for the same Qwen3-4B-Base and Qwen3-1.7B-Base runs, following the protocol of the VLM experiments (Appendix B.3) and averaging the per-benchmark consistent fractions equally over the six mathematical-reasoning benchmarks. Figure 13 reports the result. Unlike in our 7B VLM RLVR experiments, where all trained policies underperform the pre-RLVR model (Figure 7), all RLVR variants here improve consistency over the pre-RL model at both model sizes. In the text-only setting as well, PEAK improves consistency over both DAPO baselines at both scales, reaching 81.04% at 4B and 76.73% at 1.7B, with the larger margin over the baseline without KL regularization. As with final-answer

accuracy, the comparison with KL-Cov, which instead prevents entropy collapse, depends on model size: KL-Cov is more consistent at 4B, whereas PEAK is more consistent at 1.7B.

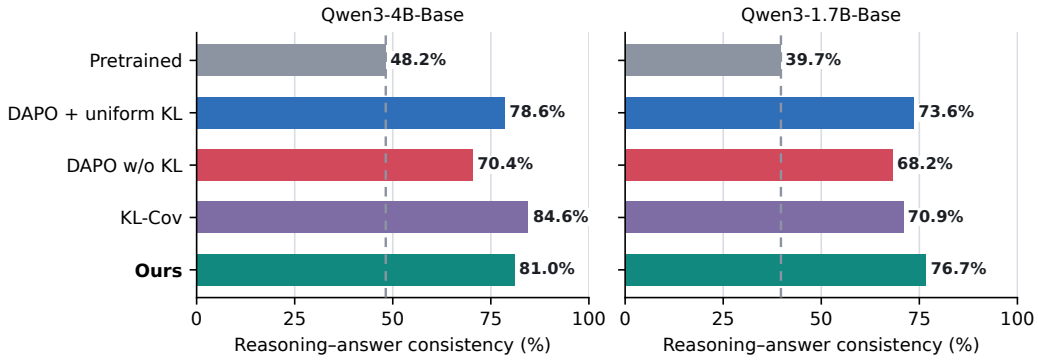


Figure 13: **Reasoning–answer consistency in the text-only setting.** GPT-OSS-120B judgments averaged equally over the six mathematical-reasoning benchmarks, following the protocol of the VLM experiments. All trained methods use DAPO. The dashed line marks the pretrained model. Blue denotes uniform KL, red no KL, purple KL-Cov, and teal PEAK.

E LIMITATIONS

This work investigates reduced reliance on generated reasoning in VLM RLVR and introduces PEAK to mitigate the associated degradation in reasoning behavior. Our intervention experiments support this connection but do not establish the causal mechanism linking RLVR, reduced reasoning reliance, and reasoning–answer inconsistency. Moreover, in our main 7B experiments, consistency and Pass@k remain below the pre-RLVR levels, indicating that PEAK only partially addresses this degradation.

In text-only LLMs, PEAK also improves over uniform-KL and no-KL baselines, but its relative performance against methods targeting entropy collapse varies with model size. This suggests an opportunity to jointly address excessive entropy growth and entropy collapse, selectively encouraging or constraining exploration according to the token context. Refining the current linear positional weighting is another direction for future work.