SAME MASTERY, OPPOSITE DIRECTIONS: SIGNED LEARNING PROGRESS FOR PROMPT REVISITING IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

While reinforcement learning with verifiable rewards has improved reasoning, the dominant cost is grouped on-policy generation, and most methods still spend a fixed group budget according to a prompt's current state. We study this allocation through learning progress, the recent change in pass rate relative to a slower mastery state. A comparison of uniformly sampled GRPO revisits uncovers two facts. First, current difficulty, competence match, and the magnitude of historical advantage assign the same priority to two prompts with the same pass rate, whether the policy is improving or regressing, while an all-correct or all-incorrect group has zero relative advantage. Second, with mastery and the size of the stored change held fixed, the next visit raises the pass rate by 22.4 points when that change is positive and by 1.1 points when it is negative, so another group is better spent on the improving prompt. Based on these observations, we propose LP-GRPO, which reads a slow mastery state and a fast signed progress state from before the current group. The state is used twice: LP-Sampling spends the next groups where a nonzero advantage is still available and progress is positive, and LP-Reweighting multiplies the resulting nonzero advantages by a positive weight from the same state. Neither step adds a selector or a value model. On eight multimodal benchmarks, with the same number of retained groups per update, LP-GRPO raises GRPO from 66.50 to 69.75 and exceeds the strongest matched baseline by 1.62 points. The revisit rule alone reaches 69.18. The full method matches GRPO's final average in half the updates, and CDAS and PSPO in two thirds.

1 INTRODUCTION

Reinforcement learning with verifiable rewards is now a standard way to improve reasoning, in language models and in vision-language models alike. The usual online algorithm is GRPO (Shao et al., 2024). It skips a learned value model, draws a group of responses for one prompt, and forms the advantage from the rewards inside that group. Those responses are the costly step. Each update retains a fixed number of groups, and every group is a fresh on-policy generation. Once that count is fixed, methods that share an optimizer and a group size differ in where the groups go, and in how strongly a nonzero advantage enters the update. The more sample-efficient rule is the one that turns the same retained groups into higher accuracy, and that reaches another method's final accuracy after fewer updates.

Under uniform traversal, many of those groups land where the advantage is zero. When every response is correct, or every response is incorrect, the rewards inside the group have no spread, and the generation does not move the policy. Leaving that traversal is a different decision from using learning direction. DAPO generates groups and then discards the ones whose rewards are uniform (Yu et al., 2025), so the discarded generation has already been paid for. CDAS keeps prompts whose estimated difficulty matches the model's competence (Kong et al., 2026). Curricula aimed at intermediate difficulty, including PCL, rank a prompt by a pass rate or by a predictor of that pass rate (Gao et al., 2026b). PSPO orders prompts by the magnitude of historical advantages and smooths prompt-level rewards for the advantage itself (Zhu et al., 2026). Each of these scores marks a position, so an improving prompt and a regressing one can share the next group's priority.

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.

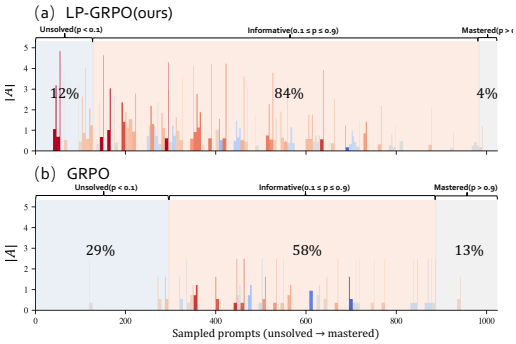


Figure 1: One step's rollout budget for (a) LP-GRPO and (b) GRPO. Each bar is a response ordered from unsolved to mastered by its pre-step historical mastery; bar height is the advantage magnitude $|A|$ and color the learning progress.

Figure 1 lines one update up from nearly unsolved to mastered. GRPO places 29% of its groups on nearly unsolved prompts and 13% on mastered ones, 42% at the two extremes, with 58% in the band that can still mix successes and failures. LP-GRPO places 12%, 84%, and 4% on those three bands. The same number of groups is kept. What changes is which prompts receive them. The middle band is where a group of fixed size is most likely to contain both a correct and an incorrect response, and where the relative advantage is therefore available to the update. Directing groups toward this informative region is the core motivation of our prompt allocation strategy.

Learning progress is the recent change around a slower mastery state. Mastery is a smoothed pass rate and moves only a little on each visit. Progress smooths the residual between the new pass rate and that mastery more quickly, so its sign is the direction of the latest visits. On 14,795 revisits from uniform GRPO, and using only the state stored before the next visit, 142 strata hold mastery, the absolute value of progress, the training stage, and the wait fixed. Inside a stratum the next visit raises the pass rate by 22.4 points when the stored progress is positive and by 1.1 points when it is negative. The current pass rate is the same; the prompt that is improving is the one that repays another group. This large performance gap demonstrates that static difficulty alone cannot capture the model's dynamic learning behavior.

LP-GRPO spends the fixed group budget with that comparison. Each prompt stores two scalars, written after the policy update and read on the next visit, so the group just drawn is scored with earlier state. The residual is the new pass rate minus stored mastery. Progress averages that residual quickly; mastery takes a smaller step along it. A first visit has progress zero, and the score falls back to mastery. LP-Sampling is the main use of the state. Its score multiplies two terms: a signal potential that is highest near a pass rate of one half and falls away as mastery approaches zero or one, and a factor above one only when stored progress is positive. Stalled and regressing prompts keep the mastery term and stay in the pool. LP-Reweighting then multiplies the GRPO advantages by a positive weight from the same stored state, so an all-correct or all-incorrect group stays at advantage zero.

We evaluate the allocation on eight multimodal reasoning benchmarks with Qwen3-VL-8B-Instruct (Bai et al., 2025). Every method trained for the comparison retains the same number of groups at each update. LP-GRPO moves the average from GRPO's 66.50 to 69.75, 1.62 points above CDAS, the strongest matched baseline. LP-Sampling, with the revisit rule alone, reaches 69.18, so most of the gain is in where the groups are spent. Ranking by mastery alone yields 68.29, and boosting negative progress yields 66.89, on every benchmark. GRPO's final average appears after half the updates, and the final averages of CDAS and PSPO after two thirds. The same rule on Qwen2.5-VL-7B reaches 58.20, 5.10 above that suite's GRPO and 0.70 above VPPO. In both settings the next group goes to the prompt whose stored progress is positive. These cross-model results verify that our learning-progress paradigm generalizes across different model sizes.

Our contributions are threefold:

- We treat sample efficiency in grouped RLVR as a revisit decision. On uniformly sampled GRPO, prompts at the same mastery split by the sign of stored progress: the next visit gains 22.4 pass-rate points when the sign is positive and 1.1 points when it is negative.
- We introduce LP-GRPO, which keeps a slow mastery state and a fast signed progress state and reads both from before the current group. LP-Sampling spends the fixed budget on prompts that can still yield a nonzero advantage and whose progress is positive. LP-Reweighting scales the resulting nonzero advantages with the same state.
- On eight benchmarks and the same group budget, LP-GRPO improves GRPO by 3.25 points and the strongest matched baseline by 1.62, and reaches that accuracy in fewer updates.

2 Related Work

Where the group budget goes. GRPO estimates relative advantages from multiple responses to the same prompt, avoiding a separate value model but making online generation a dominant cost (Shao et al., 2024). DAPO oversamples and then drops groups whose rewards are uniform, after that generation has been spent (Yu et al., 2025). CDAS estimates difficulty from historical performance and samples prompts whose difficulty matches the model's competence (Kong et al., 2026). That history stabilizes where a prompt sits, so an improving prompt and a regressing prompt at the same mastery still share a priority. Other schedulers select an intermediate band with a learned predictor or a small predictive model, or allocate the rollout budget from gradient statistics and predicted gradient variance (Gao et al., 2026b; Qu et al., 2026; Zeng et al., 2026; Nguyen et al., 2026). Further work reweights difficulty groups or evolves the training pool from difficulty (Panaganti et al., 2026; Zhang et al., 2026). LP-GRPO keeps the retained-group budget fixed and spends the next group on the prompt whose stored progress is positive.

Multimodal checkpoints and advantage design. Grouped RL with verifiable or step-wise rewards is already used for visual reasoning (Liu et al., 2025b; Zhang et al., 2025). The public models in our table continue that line (Meng et al., 2025; Wang et al., 2025b;a; Liu et al., 2025a; Yao et al., 2025; Wang et al., 2025c; Sarch et al., 2026; Feng et al., 2025; UCSC-VLAA, 2026). Closer to the update itself, multimodal difficulty sampling and task curricula organize data by difficulty (Qi et al., 2026; Yan et al., 2026), DIVA-GRPO reshapes advantages with difficulty-adaptive variants, and VPPO reweights tokens by visual dependence (Gao et al., 2026a; Huang et al., 2026). These methods change the data, the reward, or the token weight. LP-GRPO changes which prompt receives the next group.

History and learning progress. PSPO orders prompts by the magnitude of historical advantages and smooths prompt-level rewards for the advantage estimate (Zhu et al., 2026). The same interest in what is worth revisiting appears in prompt replay, learning-progress curricula, staged positive and negative advantages, and fast–slow prioritization of tasks (Baroian & Berger, 2026; Graves et al., 2017; Yang et al., 2026; Zhang et al., 2024a; Jiang et al., 2021). LP-GRPO turns that history into a slow mastery state and a fast signed progress state. LP-Sampling uses the stored state to choose the next revisit, and LP-Reweighting uses it to set the strength of the update. The current group's outcomes are written back only after that update.

3 Method

Overview. LP-GRPO treats prompt selection as a revisit decision. As shown in Figure 2, it maintains two scalars for each prompt: a slow historical-mastery state and a fast signed learning-progress state. **LP-Sampling** uses the stored state to decide where future rollout groups are spent, and **LP-Reweighting** uses the same pre-rollout state to control how strongly the selected groups update the policy. At update t, both operations read only state available through $t-1$; current outcomes are written back after the policy update and can affect only later visits.

Figure 2 reads only the state stored before the current group. LP-Sampling turns that state into the next batch. LP-Reweighting then scales the nonzero advantages, and the pass rate from this visit is written back after the update.

3.1 Two-Timescale Prompt State

For each prompt x, initial mastery is a pass rate taken before training or from the first online visit. Learning progress starts at zero, and the same update applies in either case. When the prompt is selected at update t, its G online responses provide the current pass rate:

$$p_0^x = \frac{1}{G_0}\sum_{i=1}^{G_0}\mathbb{I}[r_{0,i}^x > \tau], \tag{1}$$

$$p_t^x = \frac{1}{G}\sum_{i=1}^{G}\mathbb{I}[r_{t,i}^x > \tau]. \tag{2}$$

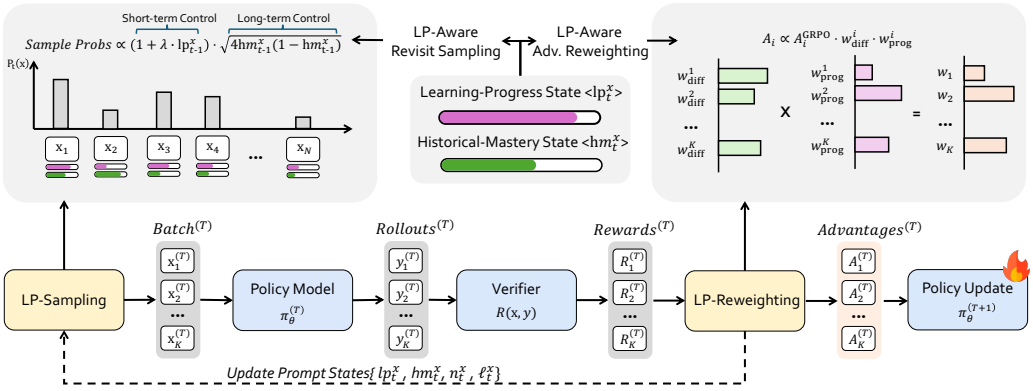


Figure 2: Overview of LP-GRPO. Each prompt keeps a slow mastery state and a fast signed progress state from the previous visit. LP-Sampling uses that state to choose which prompts receive the next groups. After the group is generated, LP-Reweighting scales the nonzero advantages with the same state. The new pass rate is written back only after the policy update.

Here, τ is the pass threshold. We set hm_0^x from the chosen initial pass rate and $\mathrm{lp}_0^x = 0$. When x is visited at step t, we compute its mastery residual and update the two states at different timescales:

$$d_t^x = p_t^x - \mathrm{hm}_{t-1}^x, \tag{3}$$

$$\mathrm{lp}_t^x = (1-\alpha)\mathrm{lp}_{t-1}^x + \alpha d_t^x, \tag{4}$$

$$\mathrm{hm}_t^x = \mathrm{hm}_{t-1}^x + \beta d_t^x. \tag{5}$$

Here, $0 < \beta < \alpha < 1$. The slow state hm_t^x summarizes historical mastery, whereas the fast state lp_t^x tracks recent deviations around that reference. Positive, near-zero, and negative values indicate improvement, stability, and regression, respectively. The state is updated only when x is visited, so prompts with similar mastery may retain different recent directions.

3.2 LP-Sampling: Revisiting Improving Prompts

LP-Sampling separates *eligibility* from *priority*. Every fresh prompt is eligible; a previously visited prompt becomes eligible again after a minimum interval I, provided that it has been visited fewer than M times. We denote the resulting candidate set by $\mathcal{E}_t$. Thus, eligibility controls *when* a prompt may return, while the stored state determines *which* eligible prompt is preferred.

Current mastery first estimates whether another group can provide a relative training signal. For a prompt with binary success probability p, the probability that G responses contain both a success and a failure is

$$\pi_G(p) = 1 - p^G - (1-p)^G. \tag{6}$$

For fixed G, this probability has the same ranking as the normalized Bernoulli standard deviation. We therefore use the bounded signal-potential score

$$g_t^x = \sqrt{4\,\mathrm{hm}_{t-1}^x(1-\mathrm{hm}_{t-1}^x)}. \tag{7}$$

It is largest at intermediate mastery and vanishes at the two extremes. Stored learning progress then distinguishes prompts at a similar mastery level:

$$s_t^x = \big(1 + \lambda[\mathrm{lp}_{t-1}^x]_+\big) g_t^x. \tag{8}$$

Here, $[z]_+ = \max(z, 0)$ and $\lambda \geq 0$. Positive progress receives an additional revisit boost; stalled and regressing prompts retain the mastery-based score rather than being discarded, preserving opportunities for exploration and recovery. Fresh prompts use $\mathrm{hm}_{t-1}^x = p_0^x$ and $\mathrm{lp}_{t-1}^x = 0$.

After applying small coverage safeguards for fresh and zero-prior prompts, LP-Sampling draws sequentially without replacement. If $\mathcal{R}$ is the set of candidates remaining in the current batch and $\bar{s}_t^x$ is the safeguarded score, then

$$P_t(x \mid \mathcal{R}) = \frac{\bar{s}_t^x}{\sum_{z\in\mathcal{R}} \bar{s}_t^z} \qquad x \in \mathcal{R}. \tag{9}$$

The exact safeguards and the equivalent visit-count mask are in the supplement.

3.3 LP-REWEIGHTING: CONTROLLING UPDATE STRENGTH

Sampling controls how often a prompt is trained; reweighting controls its contribution once selected. Crucially, the ICLR formulation uses only the progress state stored before generation. Define

$$\mathrm{lp}^x_{t,\mathrm{rw}} = \begin{cases} \mathrm{lp}^x_{t-1}, & x \text{ has been visited,} \\ 0, & x \text{ is fresh.} \end{cases} \tag{10}$$

Fresh prompts are therefore neutral with respect to progress; their current outcomes affect reweighting only on a future revisit. Let $\bar{p}^x_0$ be the initial pass rate, from either initialization above, clipped away from zero and one. The initial-difficulty term is

$$w^x_{\mathrm{diff}} = (1 - \bar{p}^x_0)^\gamma. \tag{11}$$

The bounded asymmetric progress term is

$$w^x_{\mathrm{prog},t} = \begin{cases} 1 + \alpha_p \tanh(\kappa\, \mathrm{lp}^x_{t,\mathrm{rw}}), & \mathrm{lp}^x_{t,\mathrm{rw}} \geq 0, \\ 1 + \beta_p \tanh(\kappa\, |\mathrm{lp}^x_{t,\mathrm{rw}}|), & \mathrm{lp}^x_{t,\mathrm{rw}} < 0. \end{cases} \tag{12}$$

Here, $0 < \beta_p < \alpha_p$, so equal-magnitude positive progress receives a larger multiplier than negative progress, while regressing prompts retain a bounded corrective contribution. The total prompt weight and weighted advantage are

$$w^x_t = w^x_{\mathrm{diff}} w^x_{\mathrm{prog},t}, \tag{13}$$

$$\widetilde{A}^x_{i,t} = w^x_t A^{x,\mathrm{GRPO}}_{i,t}. \tag{14}$$

Since $w^x_t > 0$, reweighting preserves the sign of every GRPO advantage and cannot create a signal for a reward-homogeneous group. The weight is an unnormalized prompt-level objective weight: it changes both relative prompt contributions and the batch's aggregate update scale. We retain this definition rather than describing it as a learning-rate change, and separately compare constant-scale, difficulty-only, progress-only, and combined controls in the evaluation. Apart from replacing $A^{x,\mathrm{GRPO}}_{i,t}$ with $\widetilde{A}^x_{i,t}$, the clipped GRPO objective is unchanged.

Update order. At each update, LP-GRPO (1) constructs the eligible set and samples prompts from state stored through $t-1$; (2) generates and scores G responses per prompt; (3) computes standard GRPO advantages and applies the stored-state multiplier; (4) updates the policy; and (5) writes the current pass rates back to Eqs. 3–5. This ordering keeps both prompt selection and progress reweighting prospective: a group's outcome cannot raise either its own sampling probability or its own progress factor. LP-GRPO stores only a constant number of scalars per prompt and requires no auxiliary selector or additional online rollout.

4 EXPERIMENTS

We organize the evaluation around three questions. **RQ1** asks whether LP-GRPO improves endpoint accuracy under a matched retained-group budget, and whether that accuracy arrives at an earlier step. **RQ2** isolates generic revisiting, current mastery, and temporal direction. **RQ3** reports how pass rate moves from the first visit through the fourth. We then report revisit coverage through step 360 and a text-only protocol.

4.1 EXPERIMENTAL SETUP

Model and training data. We use Qwen3-VL-8B-Instruct (Bai et al., 2025) as the base model. Training prompts are drawn from the MMRL source released with ReVisual-R1 (Chen et al., 2025) and the ViRL39K source released with VPPO (Huang et al., 2026). After cross-source deduplication and filtering, the resulting pool contains 30,445 prompts. All RL methods trained by us start from the same base model and use the same prompt pool; full construction details are provided in the supplementary material.

Method	DynaMath	LogicVista	MathVerse	MathVision	MathVista	MMK12	MMMU-Pro	WeMath	**Avg.**
Public 7B RL Models									
MM-Eureka-7B†	65.4	46.3	67.1	31.1	-	67.5	30.3	65.5	51.7
ThinkLite-7B†	64.6	39.4	64.2	32.0	-	62.6	28.0	66.5	49.4
VL-Rethinker-7B†	65.7	46.3	68.8	31.9	-	68.3	37.0	68.9	53.5
NoisyRollout-7B†	65.5	47.3	67.8	22.1	-	50.0	34.5	71.0	51.3
R1-ShareVL-7B†	65.1	45.6	68.0	30.1	-	70.9	35.1	69.9	53.2
PAPO-D-7B	66.8†	46.7	68.6	30.6†	-	80.6	36.3	68.3	55.3
Qwen2.5-VL-7B									
Qwen2.5-VL-7B	55.7	42.4	39.0	18.4	-	42.5	25.1	46.4	38.3
+ GRPO	65.8	45.6	66.5	30.7	-	72.3	35.2	68.1	53.1
+ DAPO	66.6	46.8	68.3	30.5	-	82.1	35.9	68.0	55.0
+ VPPO	68.1	47.9	71.6	33.3	-	82.8	37.9	71.5	57.5
+ LP-GRPO (ours)	**68.8**	**48.5**	**72.4**	**34.0**	-	**83.6**	**39.0**	**72.2**	**58.20**
Public Qwen3-VL-8B RL Checkpoints									
Vero-Qwen3I-8B†	68.90	62.42	**70.81**	60.59	**80.90**	78.43	59.94	66.00	68.50
OneThinker-8B†	67.70	55.93	66.83	48.98	76.50	71.83	54.34	59.81	62.74
VLM-CapCurriculum-Staged†	69.60	59.28	67.72	53.65	79.10	75.13	54.97	61.33	65.10
Qwen3-VL-8B-Instruct with Matched Training									
Qwen3-VL-8B	67.70	55.30	62.10	53.90	77.20	67.00	55.90	55.43	61.82
+ GRPO	70.80	62.42	69.85	54.05	78.10	77.66	59.13	60.00	66.50
+ DAPO	70.64	58.61	67.46	52.99	78.50	76.95	57.63	65.14	65.99
+ CDAS	70.56	60.18	68.12	**61.35**	79.60	**80.56**	59.25	65.43	68.13
+ VPPO	72.24	60.63	70.23	57.73	80.40	78.38	58.84	65.14	67.95
+ DIVA-GRPO	70.08	62.42	70.42	58.06	77.10	78.88	59.60	64.38	67.62
+ LP-GRPO (ours)	**73.21**	**64.43**	**70.81**	**61.35**	80.30	79.19	**61.68**	**67.05**	**69.75**

Table 1: Multimodal accuracy (%). LP-GRPO reaches 58.20 on Qwen2.5-VL-7B and 69.75 on Qwen3-VL-8B-Instruct. The 7B numbers follow Huang et al. (2026), with our LP-GRPO added. The 8B rows share the base model, data, group size, and step budget. † marks released checkpoints (Sarch et al., 2026; Feng et al., 2025; UCSC-VLAA, 2026). Bold and underline are the best and second-best in each block.

Baselines and training configuration. We compare LP-GRPO with GRPO, DAPO (Yu et al., 2025), CDAS (Kong et al., 2026), PSPO (Zhu et al., 2026), VPPO (Huang et al., 2026), and DIVA-GRPO (Gao et al., 2026a). Every method generates eight rollouts per prompt, retains 128 prompt groups per step, and uses a learning rate of 1×10^{-5}. All matched comparisons use step 360, corresponding to 46,080 prompt groups retained for policy optimization, irrespective of any longer configured training schedule. For LP-GRPO, we use $(\alpha, \beta) = (0.4, 0.2)$, $\lambda = 1$, $\gamma = 0.5$, $(\alpha_p, \beta_p) = (1.0, 0.4)$, $\kappa = 8$, $I = 30$, and $M = 5$; complete parameter settings and implementation details are provided in the supplementary material.

Evaluation protocol. We use VLMEvalKit (Duan et al., 2024) with deterministic decoding at temperature zero and report accuracy on DynaMath (Zou et al., 2024), LogicVista (Xiao et al., 2024), MathVerse (Zhang et al., 2024b), MathVision (Wang et al., 2024), MathVista (Lu et al., 2024), MMK12 (Meng et al., 2025), MMMU-Pro (Yue et al., 2024), and WeMath (Qiao et al., 2024). The aggregate score is their unweighted mean.

4.2 RQ1: Matched-Budget Endpoint Accuracy

In Table 1, LP-GRPO reaches 58.20 on Qwen2.5-VL-7B, 5.10 above GRPO and 0.70 above VPPO, and 69.75 on Qwen3-VL-8B, 3.25 above GRPO and 1.62 above CDAS, under the matched retained-group setup. LP-Sampling alone reaches 69.18, so the revisit rule accounts for most of the gain (Table 2). The gain shows up across the eight benchmarks.

LP-GRPO improves over GRPO on all eight benchmarks. The largest gains are on MathVision and WeMath, 7.30 and 7.05 points. LP-Sampling is strongest on DynaMath and MathVerse; adding reweighting brings the average to 69.75.

Figure 3 plots the same runs. Every method retains 128 groups per step, so the horizontal axis is the online budget. LP-Sampling is slightly higher at step 120; the full method leads thereafter.

Method	DynaMath	LogicVista	MathVerse	MathVision	MathVista	MMK12	MMMU-Pro	WeMath	**Avg.**
+ GRPO	70.80	62.42	69.85	54.05	78.10	77.66	59.13	60.00	66.50
+ LP-Reweighting	71.80	59.73	68.38	56.64	79.00	**80.00**	58.38	**67.24**	67.65
+ LP-Sampling	**73.71**	62.42	**70.86**	59.05	**81.30**	79.64	61.39	65.05	69.18
+ LP-GRPO (ours)	73.21	**64.43**	70.81	**61.35**	80.30	79.19	**61.68**	67.05	**69.75**

Table 2: Ablation on Qwen3-VL-8B-Instruct under the matched retained-group budget. LP-Reweighting keeps uniform sampling and scales advantages. LP-Sampling changes only the revisit rule. Bold and underline are the best and second-best in each column.

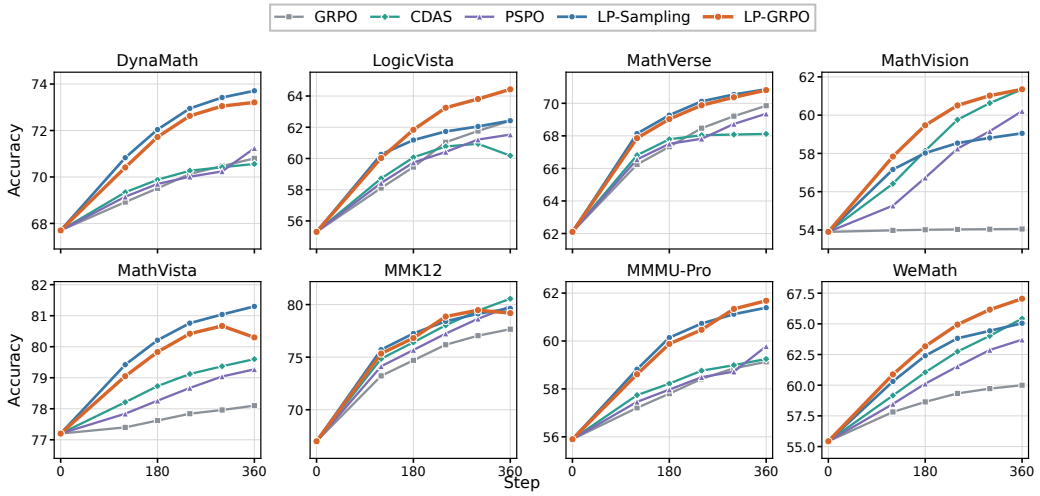


Figure 3: Accuracy (%) versus training step on Qwen3-VL-8B-Instruct. Each panel is one of the eight benchmarks. Every method retains 128 groups per step, so the horizontal axis is the matched online budget. Markers are checkpoints at steps $\{0, 120, 180, 240, 300, 360\}$. LP-Sampling is slightly ahead at step 120; LP-GRPO leads thereafter and finishes at a 69.75 eight-benchmark average against GRPO at 66.50.

Cost efficiency. Every method retains 128 groups per step. LP-GRPO passes GRPO's step-360 average of 66.50 at step 180, where it already reaches 67.72, halfway through the matched schedule. It passes CDAS (68.13) and PSPO (68.12) at step 240, where it reaches 68.87, two thirds of the way. LP-Sampling is already at 67.56 and 68.38 at those two steps.

Figure 4 follows GRPO and LP-GRPO through step 360. LP-GRPO's reward rises from 0.594 to 0.684, against GRPO's 0.473 to 0.542. Over the last 50 steps, 55.7% of LP-GRPO's groups are still improving, against 42.5% for GRPO.

4.3 RQ2: What Does Temporal Direction Add Beyond Mastery?

Figure 5 compares three sampling rules that share the same eligibility constraint, batch size, and optimizer. Positive LP is the LP-Sampling run. No progress removes the progress boost ($\lambda = 0$) and ranks prompts by mastery only. Negative LP boosts regressing prompts instead. On every benchmark and at every reported checkpoint, the order is positive, then no progress, then negative. At step 360 the eight-benchmark averages are 69.18, 68.29, and 66.89. Progress therefore adds accuracy beyond mastery, and the useful direction is improvement rather than regression.

4.4 RQ3: How Does Pass Rate Move Across Revisits?

Table 3 follows prompts that receive later visits and groups them by successes on the first visit. Hard prompts rise from 0 to 0.219 and learnable prompts from 0.265 to 0.501, gains of 21.9 and 23.6

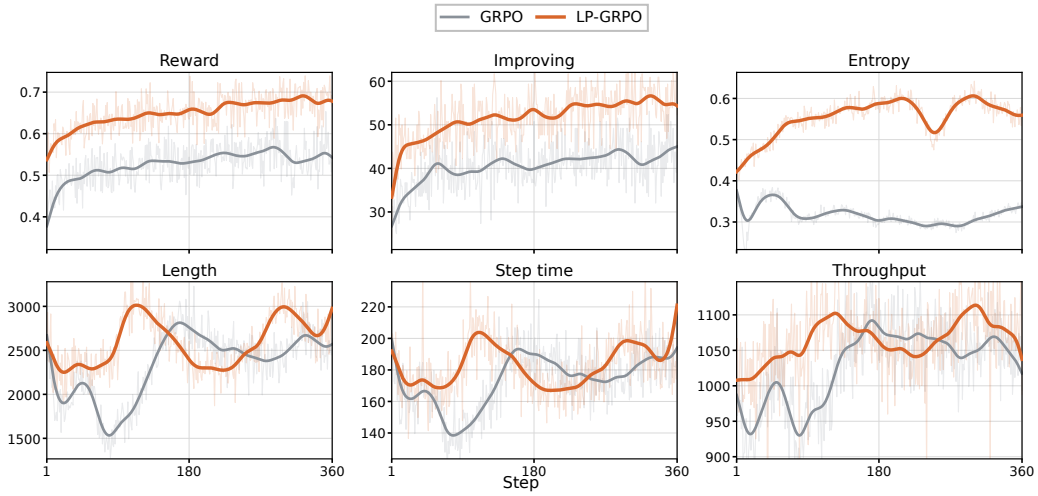


Figure 4: Training dynamics of GRPO and LP-GRPO through step 360. Faint curves are the per-step log values and bold curves are Gaussian smooths. Panels report sampled-group reward, the share of improving groups, entropy, response length, step time, and throughput. LP-GRPO's reward rises from 0.594 to 0.684 against GRPO's 0.473 to 0.542, and over the last 50 steps 55.7% of its groups are still improving against 42.5% for GRPO.

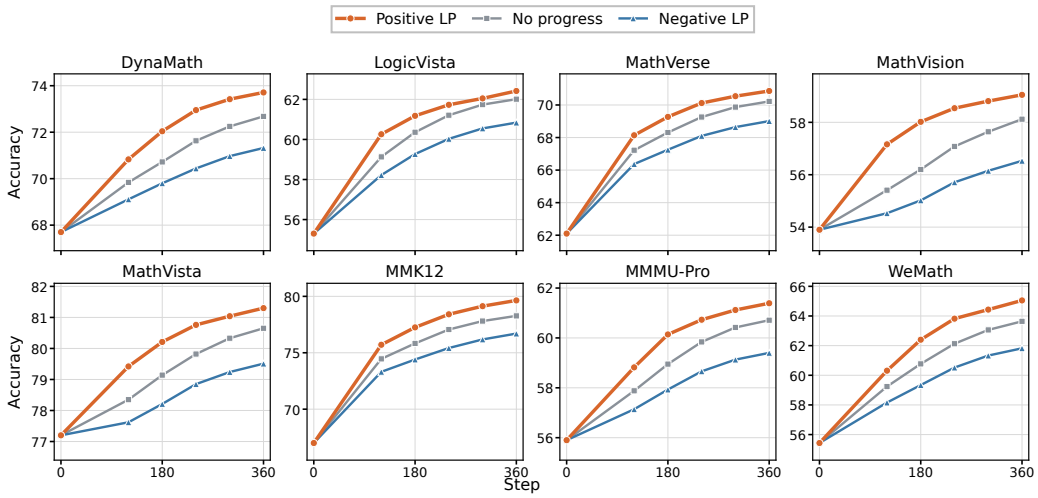


Figure 5: Positive, no-progress, and negative sampling on Qwen3-VL-8B-Instruct under the matched retained-group budget. Each panel is one benchmark. Positive LP is the LP-Sampling run; no progress removes the progress boost ($\lambda = 0$) and ranks by mastery only; negative LP boosts regressing prompts. The order is positive, then no progress, then negative on every benchmark, with step-360 eight-benchmark averages 69.18, 68.29, and 66.89.

points by the fourth visit. Mixed prompts gain 13.1 points. Easy prompts stay near 0.92. The mean advantage multiplier is 1.86 on hard prompts and 0.32 on easy prompts; GRPO's weight stays at 1.

4.5 Revisit Coverage

Table 4 follows the same matched run from the pool to the optimizer. At step 360, LP-Sampling has covered 77.66% of the prompt pool. The fresh-prompt share falls from 96.09% at step 40 to 20.31%, and prompts already seen average 1.96 visits. Selected prompts score 0.853, against 0.681 for the

	Hard 0/8	Learnable 1–3/8	Mixed 4–5/8	Easy 6–8/8
V1	.000	.265	.575	.924
V2	.149	.418	.639	.908
V3	.171	.451	.678	.915
V4	.219	.501	.706	.920
$\Delta_{1\to4}$	+21.9	**+23.6**	+13.1	-0.4
$\bar{w}$	**1.86**	1.53	0.93	0.32

Table 3: Pass rate from the first visit to the fourth. Columns are grouped by successes on the first visit. $\Delta_{1\to4}$ is in percentage points, and $\bar{w}$ is the mean advantage multiplier.

Step	Coverage	Fresh	Visits	Sel.	Elig.	Stored LP	Mean w
40	16.80	96.09	1.01	0.732	0.636	5	1.091
120	42.33	66.41	1.20	0.828	0.652	43	1.096
240	65.30	28.13	1.55	0.870	0.671	92	1.154
360	77.66	20.31	1.96	0.853	0.681	102	1.067

Table 4: LP-Sampling and LP-Reweighting over 360 training steps. Coverage and fresh-prompt share are percentages. Visits denotes the average visit count for seen prompts. Sel. and Elig. are mean sampling scores of the selected batch and eligible pool. Stored LP counts groups (out of 128) using progress from prior visits for advantage weighting.

eligible pool. Groups whose advantage weight reuses progress stored on an earlier visit rise from 5 of 128 at step 40 to 102 at step 360. The mean weight stays between 1.067 and 1.154.

4.6 Text-Only Protocol

Table 5 applies the same method on the Qwen2.5-7B, 55-step CDAS text protocol. LP-GRPO reaches 44.86, above CDAS at 44.51, and above prioritized sampling at 44.06. Random sampling and curriculum sampling both finish at 43.20. LP-GRPO is highest on MATH500 (77.12), Minerva (39.46), and GSM8K (92.85). The PSPO row is the published Qwen2.5-Math-7B result and is not pooled with that average.

Method	AIME24	AIME25	MATH500	Minerva	Olympiad	GSM8K	**Avg.**
Random Sampling	10.00	7.29	74.20	36.76	39.26	91.66	43.20
Curriculum Sampling	10.00	7.29	74.20	36.76	39.26	91.66	43.20
Prioritized Sampling	11.77	7.08	74.00	39.34	40.15	92.04	44.06
Dynamic Sampling	**12.19**	**7.92**	75.00	38.97	38.52	91.36	43.99
CDAS	11.56	7.71	76.20	38.97	**40.44**	92.19	44.51
PSPO	10.00	–	58.40	14.70	24.40	–	30.50
LP-GRPO (ours)	12.08	7.53	**77.12**	**39.46**	40.12	**92.85**	**44.86**

Table 5: Text-only. Qwen2.5-7B, 55-step CDAS protocol, except PSPO, which is the published Qwen2.5-Math-7B result (Zhu et al., 2026).

5 Conclusion

LP-GRPO treats prompt allocation as a decision combining available training signals and stored learning direction. With fixed group budgets, LP-GRPO surpasses GRPO by 3.25 points and the strongest baseline by 1.62 over eight multimodal benchmarks. It reaches GRPO's step-360 performance at step 180 and CDAS/PSPO's final accuracy at step 240. Sampling drives most gains: positive-progress sampling outperforms mastery-only and negative-progress variants, with the largest improvements on hard yet learnable prompts. At step 360, the sampler covers 77.66% of the prompt pool, fresh prompts drop to 20.31%, and our method attains 44.86 on the text protocol.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability, to organize the literature search. We have not used generative AI tools to generate synthetic data, develop theoretical claims or proofs, propose hypotheses, design the experiments, or create or alter experimental measurements. Survey instruments and transcription of recordings are not applicable to this work. We reviewed all AI-assisted text, and independently verified the technical claims, citations, and reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The update order, safeguards, and hyperparameters are documented in the appendix, together with the data construction, hardware, and seeds. Per-benchmark results are reported in Section 4 and the appendix. The accompanying artifact contains the paper sources, implementation snapshots, resolved configurations, and analysis scripts. External model weights, datasets, and benchmark packages are referenced and are not redistributed.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, et al. Qwen3-VL technical report, 2025.

Andrei Baroian and Rutger Berger. Prompt replay: Speeding up GRPO with on-policy reuse of high-signal prompts, 2026.

Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning, 2025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, et al. OneThinker: All-in-one reasoning model for image and video, 2025.

Haowen Gao, Zhenyu Zhang, Liang Pang, Fangda Guo, Hongjian Dou, Guannan Lv, Shaoguo Liu, Tingting Gao, Huawei Shen, and Xueqi Cheng. DIVA-GRPO: Enhancing multimodal reasoning through difficulty-adaptive variant advantage. In *International Conference on Learning Representations*, 2026a.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient LLM post-training. In *International Conference on Learning Representations*, 2026b.

Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *International Conference on Machine Learning*, pp. 1311–1320, 2017.

Siyuan Huang, Xiaoye Qu, Yafu Li, Yun Luo, Zefeng He, Daizong Liu, and Yu Cheng. Spotlight on token perception for multimodal reinforcement learning. In *International Conference on Learning Representations*, 2026.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950, 2021.

Deyang Kong, Qi Guo, Xiangyu Xi, Wei Wang, Jingang Wang, Xunliang Cai, Shikun Zhang, and Wei Ye. Rethinking the sampling criteria in reinforcement learning for LLM reasoning: A competence-difficulty alignment perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31438–31446, 2026. doi: 10.1609/aaai.v40i37.40408.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. NoisyRollout: Reinforcing visual reasoning with data augmentation. *arXiv preprint arXiv:2504.13055*, 2025a.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

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

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

Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *International Conference on Learning Representations*, 2026.

Kishan Panaganti, Zhenwen Liang, Wenhao Yu, Haitao Mi, and Dong Yu. Group distributionally robust optimization-driven reinforcement learning for LLM reasoning, 2026.

Jianyu Qi, Ding Zou, Wenrui Yan, Rui Ma, Jiaxu Li, Zhijie Zheng, Zhiguo Yang, and Rongchang Zhao. Revisiting the data sampling in multimodal post-training from a difficulty-distinguish view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24891–24899, 2026. doi: 10.1609/aaai.v40i29.39676.

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

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, et al. Small generalizable prompt predictive models can steer efficient rl post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026.

Gabriel Sarch, Linrong Cai, Qunzhong Wang, Haoyang Wu, Danqi Chen, and Zhuang Liu. Vero: An open RL recipe for general visual reasoning, 2026.

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

UCSC-VLAA. VLM-CapCurriculum-Qwen3-VL-8B-Staged. Hugging Face model card, 2026. URL https://huggingface.co/UCSC-VLAA/VLM-CapCurriculum-Qwen3-VL-8B-Staged. Accessed 2026-07-14.

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. *arXiv preprint arXiv:2504.08837*, 2025a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95095–95169, 2024.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. SOTA with less: MCTS-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025b.

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. *arXiv preprint arXiv:2507.06448*, 2025c.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. LogicVista: Multimodal LLM logical reasoning benchmark in visual contexts, 2024.

Qingyang Yan, Guangyao Chen, and Yixiong Zou. Start small, think big: Curriculum-based relative policy optimization for visual grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11550–11558, 2026. doi: 10.1609/aaai.v40i14.38138.

Changpeng Yang, Jinyang Wu, Yuchen Liu, Shuai Zhang, Yang Li, Qiliang Liang, Hongzhen Wang, Shuai Nie, Jiaming Xu, Runyu Shi, Ying Huang, and Guoquan Zhang. From imitation to discrimination: Toward a generalized curriculum advantage mechanism enhancing cross-domain reasoning tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34214–34222, 2026. doi: 10.1609/aaai.v40i40.40717.

Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, et al. R1-ShareVL: Incentivizing reasoning capability of multimodal large language models via share-GRPO. *arXiv preprint arXiv:2505.16673*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale, 2025.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark, 2024.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. CurES: From gradient analysis to efficient curriculum learning for reasoning LLMs. In *International Conference on Learning Representations*, 2026.

Jenny Zhang, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. OMNI: Open-endedness via models of human notions of interestingness. In *International Conference on Learning Representations*, 2024a.

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

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

Ru Zhang, Renda Li, Ziyu Ma, Weijie Qiu, Chongyang Tao, Yong Wang, and Xiangxiang Chu. D^2Evo: Dual difficulty-aware self-evolution for data-efficient reinforcement learning. In *International Conference on Machine Learning*, 2026.

Xinxin Zhu, Ying He, Haowen Hou, Ruichong Zhang, Nianbo Zeng, Yulin Peng, Jiongfeng Fang, and F. Richard Yu. PSPO: Prompt-level prioritization and experience-weighted smoothing for efficient policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29186–29194, 2026. doi: 10.1609/aaai.v40i34.40157.

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

CONTENTS

A Data and Experimental Protocol **14**

A.1 Training Data Construction 14

A.2 Training and Evaluation Protocol 15

B Exact LP-GRPO Implementation **16**

B.1 Fixed Settings and Parameter Roles 16

B.2 Sampling Safeguards and Eligibility 16

B.3 Pass-Rate Clipping and Reweighting 16

B.4 Update Timing and Procedure 17

C Learning-Progress Positioning and Update Properties **18**

C.1 Positioning and Asymmetric Reweighting 18

C.2 Structural Properties of the LP-GRPO Update 18

D Detailed Allocation and Revisit Diagnostics **22**

D.1 Post-Hoc State Definitions 22

D.2 Allocation and Training Stability 22

D.3 Stagewise Revisit Diagnostics 23

E Hyperparameter Checks **23**

F Allocation Diagnostics **23**

G Reproducibility Scope **23**

H Qualitative Output Evolution Across Revisits **25**

H.1 Geometry 25

H.2 Graph Interpretation 25

H.3 Scientific-Plot Reading 26

A Data and Experimental Protocol

A.1 Training Data Construction

Training data construction. We profile 67,518 MMRL and ViRL39K examples; Figure 6 shows that 0/8 and 8/8 prompts dominate. Deduplication by question, answer, and image leaves 61,005 candidates. We remove 26,008 prompts at 8/8, retain all 20,445 at 1/8–7/8, and keep the 10,000 highest-entropy prompts among 14,552 at 0/8. Every method uses the resulting 30,445-prompt pool; LP-GRPO also consumes its stored p_0.

Budget accounting. Profiling produces 540,144 base-model responses across 67,518 candidates. This one-time pass constructs the common pool and supplies p_0. Each multimodal run retains 46,080 groups, or 368,640 responses.

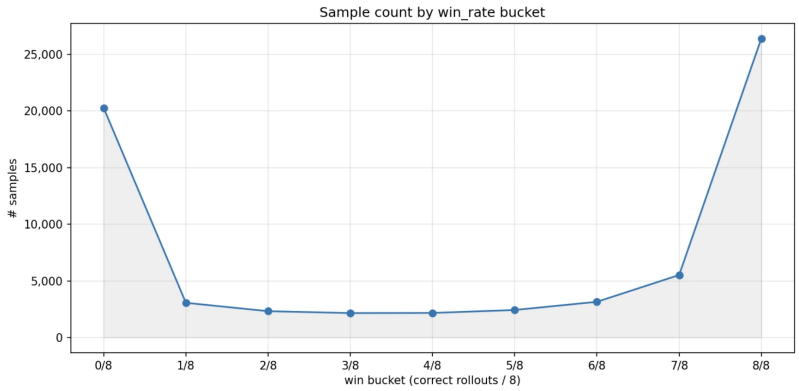


Figure 6: Number of profiled prompts in each offline win-rate bucket over eight base-model rollouts, before filtering and cross-source deduplication. The 0/8 and 8/8 endpoints account for 30.0% and 39.1% of candidates, respectively.

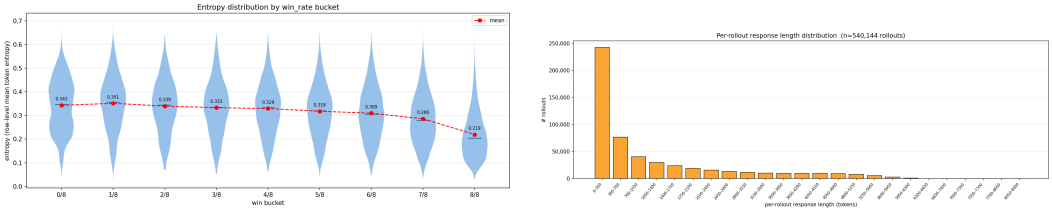


Figure 7: Prompt-pool characterization from the shared eight-rollout base-model profiling pass. (a) Base-model response entropy by win-rate bucket: mean row-level token entropy falls monotonically from 0.343 at 0/8 to 0.219 at 8/8, so the hardest prompts are also the most uncertain. (b) Per-rollout response-length distribution over all 540,144 profiling responses; most rollouts are short, with a light tail toward the length cap.

Prompt-pool characterization. Figure 7 reports two properties of the profiling pass that motivate the construction choices above. Base-model response entropy is concentrated in a moderate band—overall percentiles P10/P25/P50/P75/P90 = 0.132/0.193/0.276/0.376/0.454—and decreases monotonically with win rate. Because the 0/8 bucket is both the largest hard-prompt reservoir and the most uncertain, we retain its 10,000 highest-entropy items as the most informative hard candidates rather than a random subset. The response-length distribution is right-skewed with only a light tail near the cap, consistent with the 8,192-token response limit and dynamic batching used in training.

Shared training configuration		*LP-GRPO-specific configuration*	
Hyperparameter	Value	Hyperparameter	Value
Base model	Qwen3-VL-8B-Instruct	Progress EMA α	0.4
Training prompts	30,445	Mastery EMA β	0.2
Rollouts per prompt G	8	Reward threshold τ	0.5
Prompt groups per update	128	Minimum revisit interval I	30
Optimizer / weight decay	AdamW / 0.01	Maximum visit count M	5
Learning rate	1×10^{-5}	Progress score coefficient λ	1.0
PPO mini-batch / epochs	128 groups / 1	Score constant ϵ_s	10^{-3}
PPO clip ratio	0.2	Unseen floor ρ_{new}	0.15
KL loss / entropy coefficient	disabled / 0	Zero-prior bonus ρ_0	0.15
Dynamic batching / max tokens	enabled / 28,672	Difficulty exponent γ	0.5
Main checkpoint	update 360	Pass-rate clipping ϵ_p	10^{-3}
Maximum training schedule	3 epochs	Improving/regressing gains α_p/β_p	1.0/0.4
Maximum prompt/response length	8192/8192	Tanh sensitivity κ	8.0
Rollout engine/temperature	vLLM/1.0	Analysis-only threshold θ	0.05
Rollout top-p / top-k	1.0 / -1	LP-Sampling RNG seed	1
Actor precision/strategy	bf16/FSDP2		
Rollout tensor parallelism	4		
Training hardware count	2 nodes, 8 GPUs each		
GPU model / memory	16 NVIDIA H800 / 80 GB each		
Time to main checkpoint	18.22 h		
Actor/data-loader seed	42		
vLLM engine seed	0		

Table 6: Shared training configuration and LP-GRPO-specific hyperparameters for the multimodal runs. The reported run used 16 NVIDIA H800 GPUs (80 GB each) across two nodes. Its time to the update-360 main checkpoint is the sum of the logged training-update durations, including checkpoint writes but excluding validation and initialization. The eligible set always exceeded the batch size, so the sampler's emergency fill branch was never used.

A.2 Training and Evaluation Protocol

Training prompt and reward verifier. All matched runs share a single task-agnostic system prompt that requests an internal reasoning trace followed by a boxed final answer:

> *You FIRST think about the reasoning process step by step as an internal monologue and then provide the final answer. The reasoning process MUST BE enclosed within* `<thought> </thought>` *tags. The final answer MUST BE put in* `\boxed{}`.

The user turn appends the original benchmark question and image(s) unchanged; we add no per-benchmark instructions, few-shot exemplars, or chain-of-thought templates. The verifier is rule-based and identical for both training sources. Given a completion, it extracts the candidate answer in a fixed priority order—`\boxed{}` content first, then an `<answer>...</answer>` span, then a short list of natural-language conclusion patterns (e.g., "Therefore, the answer is")—and compares it to the ground truth with the symbolic/numeric matcher of `mathruler`. The reward is binary, 1 for a correct match and 0 otherwise, so the threshold $\tau = 0.5$ in the main paper simply realizes this pass/fail decision, and each prompt's pass rate is the mean over its G rollouts. No format-only, length, or auxiliary shaping terms are added, and the KL and entropy coefficients are zero.

Evaluation protocol. All checkpoints are evaluated with the same VLMEvalKit version and deterministic decoding. We use the official or toolkit-provided test split for each benchmark. MathVerse and MathVista use their MINI variants, and MMMU-Pro uses the 10-choice configuration. The averages in the main paper's multimodal and ablation tables are the unweighted means of the eight reported accuracies. The hyperparameter analysis changes only the displayed parameter group within each comparison and reports MATH-Vision at update 360. The repeated 61.35 entry is the same proposed checkpoint; these comparisons are post-hoc diagnostics, not a benchmark-specific model-selection procedure.

Result provenance. The multimodal rows share the base model and the prompt pool. Each trained checkpoint is from one run.

B EXACT LP-GRPO IMPLEMENTATION

B.1 FIXED SETTINGS AND PARAMETER ROLES

Parameter roles and fixed-setting rationale. The proposed configuration in Table 6 is shared across all eight multimodal benchmark evaluations; the hyperparameter checks below do not replace it on individual benchmarks. We group the method constants by function rather than treating them as a jointly tuned search space.

State dynamics. Because prompt states update only upon visits, $\alpha = 0.4$ and $\beta = 0.2$ correspond to nominal EMA horizons of $1/\alpha = 2.5$ and $1/\beta = 5$ prompt visits. Their separation makes learning progress responsive while historical mastery integrates evidence over most of the allowed five-visit history. The equal-rate check below uses $(\alpha, \beta) = (0.3, 0.3)$, preserving their mean update rate while removing the fast–slow separation. With binary correctness rewards, $\tau = 0.5$ simply implements the pass/fail decision.

Sampling and revisit control. Proposition 1 proves that initialization at $\mathrm{lp}_0^x = 0$ keeps $\mathrm{lp}_t^x \in [-1, 1]$, so $\lambda = 1$ constrains the sampling score's multiplicative progress boost to $[1, 2]$. The interval $I = 30$ prevents immediate re-selection, whereas $M = 5$ is a concentration safeguard; no prompt reaches this cap by update 360. These eligibility constants do not change the relative priority among currently eligible prompts.

B.2 SAMPLING SAFEGUARDS AND ELIGIBILITY

Coverage and numerical safeguards. For any nonempty eligible set, the score constant $\epsilon_s = 10^{-3}$ makes every eligible score positive and therefore keeps the proportional-normalization denominator nonzero. The normalized signal potential lies in $[0, 1]$; $\rho_{\mathrm{new}} = 0.15$ gives fresh prompts a modest coverage floor, while $\rho_0 = 0.15$ separately prevents offline-zero prompts from starvation. Their behavior is reported through coverage and selected-versus-eligible diagnostics rather than treated as additional performance-tuning axes. Finally, $\theta = 0.05$ is used only to summarize positive and negative progress rates in logs and never affects sampling or policy optimization.

Exact revisit safeguards. The main paper separates candidate eligibility from priority and isolates the learning-progress score from implementation safeguards. Let s_t^x be its learning-progress-modulated signal-potential score, and let n_{t-1}^x and ℓ_{t-1}^x denote the completed-visit count and most recent visit update stored before draw t. The adjusted score used by the implementation is

$$u_t^x = s_t^x + \epsilon_s \tag{15}$$

$$\widetilde{s}_t^x = \begin{cases} \max(u_t^x, \rho_{\mathrm{new}}), & n_{t-1}^x = 0 \\ u_t^x, & n_{t-1}^x > 0 \end{cases} + \rho_0 \mathbb{I}[p_0^x = 0] \tag{16}$$

$$e_t^x = \mathbb{I}[n_{t-1}^x < M]\, \mathbb{I}[t - \ell_{t-1}^x \geq I] \tag{17}$$

$$\bar{s}_t^x = e_t^x \widetilde{s}_t^x \tag{18}$$

We initialize $n_0^x = 0$ and $\ell_0^x = -\infty$, so every prompt is initially eligible. The fresh-prompt floor preserves coverage, the zero-prior bonus prevents 0/G_0 prompts from being suppressed solely by the signal-potential term, and the interval and visit cap limit immediate repetition and concentration. The mask e_t^x is the implementation form of the eligible set $\mathcal{E}_t$ defined in the main paper. If fewer positive-score candidates than the requested batch size remain, the implementation takes those candidates and emergency-fills the remaining slots from a shuffled complement. This branch was never entered through update 360 because the eligible set always exceeded the batch size.

B.3 PASS-RATE CLIPPING AND REWEIGHTING

The offline difficulty anchor uses the following explicit clipping rule:

$$\bar{p}_0^x = \min\{1 - \epsilon_p, \max\{\epsilon_p, p_0^x\}\}, \qquad \epsilon_p = 10^{-3}. \tag{19}$$

Because the offline pass rates have resolution $1/G_0 = 0.125$, Eq. (19) changes only the empirical 0/8 and 8/8 endpoints. It keeps the difficulty anchor strictly positive without changing the ordering of prompts in the interior.

Advantage-reweighting parameters. The choice $\gamma = 0.5$ gives a square-root difficulty anchor, preserving the initial-difficulty ordering while attenuating easy prompts less aggressively than a linear anchor. The gains $(\alpha_p, \beta_p) = (1.0, 0.4)$ bound the improving and regressing progress factors below 2.0 and 1.4, respectively. For equal progress magnitude $q > 0$, their exact directional gap is

$$w_{\mathrm{prog}}(q) - w_{\mathrm{prog}}(-q) = (\alpha_p - \beta_p)\tanh(\kappa q) > 0. \tag{20}$$

We set $\kappa = 8 = G$, tying the transition scale to the rollout group. For a previously neutral fast state, a one-success pass-rate change contributes $\alpha/G = 0.05$, for which $\tanh(8 \times 0.05) \approx 0.38$; the factor therefore responds without immediately saturating. The symmetric and reverse checks below hold $\alpha_p + \beta_p = 1.4$ fixed, isolating direction from total gain capacity.

B.4 Update Timing and Procedure

Implementation timing. At update t, the sampler draws prompts using only state written through update $t-1$. Advantage computation then reads the stored progress for revisited prompts; a first visit uses the current-minus-offline fallback only for advantage weighting. The policy update is computed before current outcomes are written back to the sampler. Consequently, a current group's reward cannot affect its own pre-draw selection probability, and its updated progress becomes available only to later visits.

LP-GRPO update procedure. Algorithm 1 lists the complete update order, expanding the high-level summary with the exact state reads, safeguards, and write-back timing. Every quantity stored through update $t-1$ is read before the current draw, so a group's own reward can never influence its pre-draw selection probability. Symbols follow the main paper; $\bar{s}_t^x$ and the mask e_t^x are the safeguarded score and eligibility mask defined by Eq. 15–18.

Algorithm 1: LP-GRPO training loop

Input: policy π_θ; prompt pool $\mathcal{X}$; offline pass rates $\{p_0^x\}$; rates (α, β); sampling coefficient λ; difficulty exponent γ; progress gains (α_p, β_p) and sensitivity κ; interval I, cap M; safeguards $(\epsilon_s, \rho_{\mathrm{new}}, \rho_0, \epsilon_p)$.
Init: $\mathrm{hm}_0^x = p_0^x,\ \mathrm{lp}_0^x = 0,\ n_0^x = 0,\ \ell_0^x = -\infty$ for all x.
for update $t = 1, \ldots, T$ **do**

1. *Eligibility.* Build the fresh, revisitable, and eligible sets $\mathcal{F}_t, \mathcal{V}_t, \mathcal{E}_t$ from stored n_{t-1}^x, ℓ_{t-1}^x using interval I and cap M; equivalently, form the mask e_t^x (Eq. 17).
2. *Priority.* For each $x \in \mathcal{E}_t$, read stored $\mathrm{hm}_{t-1}^x, \mathrm{lp}_{t-1}^x$; compute the signal potential, the progress-modulated score, and the safeguarded masked score $\bar{s}_t^x$ (Eq. 15–18).
3. *Draw.* Sample the batch $\mathcal{B}_t$ sequentially without replacement with probability $\propto \bar{s}_t^x$. If fewer positive-score candidates than the requested batch size remain, emergency-fill the remaining slots from a shuffled complement; this branch was not used through update 360.
4. *Rollouts.* For each $x \in \mathcal{B}_t$, generate and verify G responses; compute the current pass rate p_t^x and the original GRPO advantages $A_{i,t}^{x,\mathrm{GRPO}}$.
5. *Reweight.* Set the advantage-side progress $\mathrm{lp}_{t,\mathrm{adv}}^x$ from stored lp_{t-1}^x (revisit) or $p_t^x - p_0^x$ (first visit); form $w_t^x = w_{\mathrm{diff}}^x\, w_{\mathrm{prog},t}^x$ and the weighted advantage $\widetilde{A}_{i,t}^x = w_t^x A_{i,t}^{x,\mathrm{GRPO}}$.
6. *Optimize.* Update θ by maximizing the clipped LP-GRPO objective on $\mathcal{B}_t$ with the weighted advantages.
7. *Write back* (selected x only). Compute $d_t^x = p_t^x - \mathrm{hm}_{t-1}^x$; update $\mathrm{lp}_t^x, \mathrm{hm}_t^x$ at their two timescales; set $n_t^x = n_{t-1}^x + 1$ and $\ell_t^x = t$. Unselected states are unchanged.

end for
Output: optimized policy π_θ and updated prompt states.

C Learning-Progress Positioning and Update Properties

C.1 Positioning and Asymmetric Reweighting

Relation to broader learning-progress curricula. As reviewed in the main paper, the use of a smoothed competence change as a curriculum signal predates this work; automated curriculum methods already allocate practice toward tasks whose measured competence is changing. The learning-progress update adopts that general principle rather than claiming EMA estimation itself as new. Its role here differs in three respects. State is keyed to individual prompts and updates asynchronously only when a prompt is visited; pre-draw state schedules when that same prompt may re-enter an on-policy GRPO batch; and the same stored state subsequently scales the prompt's nonzero group-relative advantages. The contribution is this sampler–optimizer coupling under a finite revisit budget, not the generic idea of tracking learning progress.

Why positive and negative progress are asymmetric. The sign of prompt progress is not the sign of the policy gradient. Within a selected group, the original GRPO advantages still determine which responses are reinforced or suppressed. Moreover, each online pass rate is estimated from only eight Bernoulli outcomes, so a single downward change is a noisy indicator of persistent forgetting. We therefore use $0 < \beta_p < \alpha_p$ as shrinkage: regressing prompts retain a corrective gain, but a transient downward fluctuation is not amplified as strongly as repeated evidence that a prompt has entered a learnable regime. We do not claim that this asymmetry is universally optimal.

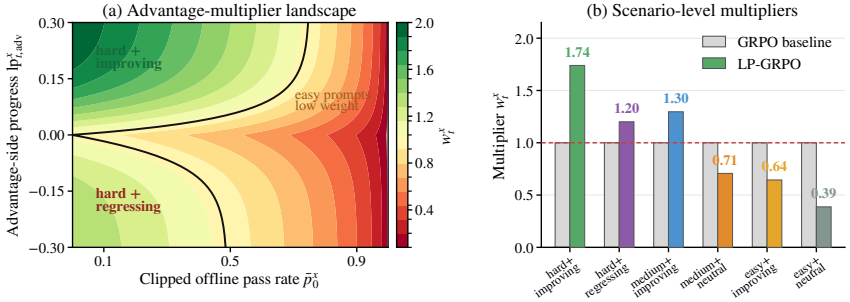


Figure 8: Deterministic visualization of the LP-GRPO advantage multiplier under the reported setting $(\epsilon_p, \gamma, \alpha_p, \beta_p, \kappa) = (10^{-3}, 0.5, 1.0, 0.4, 8)$. (a) Multiplier over the clipped offline pass rate $\bar{p}^x_0$ and advantage-side progress $\mathrm{lp}^x_{t,\mathrm{adv}}$; the black contour marks $w^x_t = 1$. (b) Calculated multipliers for illustrative hard, medium, and easy scenarios. The values visualize the weighting rule and are not empirical performance or selection-frequency measurements.

Figure 8 separates the two roles in the multiplier. The offline anchor gives initially harder prompts more weight, while the progress factor produces a larger boost for improvement than for equal-magnitude regression. Initially easy prompts can remain below unit weight even when improving because the two factors are multiplied. All displayed weights are positive, so the original GRPO advantage sign is preserved.

C.2 Structural Properties of the LP-GRPO Update

The following results isolate the estimator, sampler, and optimizer properties used by LP-GRPO. They are finite-step structural statements under explicit idealizations, not a global convergence or performance guarantee. For the state results, k counts successive visits to one fixed prompt (the state is unchanged between visits). Rewards are binary, G is the rollout-group size, and $\bar{p}_0$ is the clipped offline pass rate.

Two-timescale learning-progress state.

Proposition 1 (Invariant state bounds)**.** *Let* $p_k \in [0,1]$, $\mathrm{hm}_0 \in [0,1]$, $\mathrm{lp}_0 \in [-1,1]$, *and* $0 \le \alpha, \beta \le 1$. *Then every visit satisfies*

$$\mathrm{hm}_k \in [0,1], \qquad d_k = p_k - \mathrm{hm}_{k-1} \in [-1,1], \qquad \mathrm{lp}_k \in [-1,1]. \tag{21}$$

Consequently, for $\lambda \geq 0$, the sampling-side progress factor obeys

$$1 \leq 1 + \lambda[\mathrm{lp}_k]_+ \leq 1 + \lambda. \tag{22}$$

Proof. The mastery update can be written as

$$\mathrm{hm}_k = (1-\beta)\mathrm{hm}_{k-1} + \beta p_k, \tag{23}$$

a convex combination of two values in $[0,1]$. Hence $\mathrm{hm}_k \in [0,1]$ by induction, and therefore $d_k \in [-1,1]$. The progress update

$$\mathrm{lp}_k = (1-\alpha)\mathrm{lp}_{k-1} + \alpha d_k \tag{24}$$

is likewise a convex combination of two values in $[-1,1]$, proving (21) by induction. Applying the positive-part operator gives $[\mathrm{lp}_k]_+ \in [0,1]$, which yields (22).

Proposition 2 (Directional response to a persistent level shift)**.** *Suppose a prompt is initially at a stable success level $\mu_0 \in [0,1]$, with $\mathrm{hm}_0 = \mu_0$ and $\mathrm{lp}_0 = 0$. From visit $k = 1$ onward, suppose $\mathbb{E}[p_k] = \mu_1 = \mu_0 + \Delta \in [0,1]$, and let $0 < \beta < \alpha < 1$. Then*

$$\begin{aligned} \mathbb{E}[\mathrm{hm}_k] &= \mu_1 - \Delta(1-\beta)^k, \\ \mathbb{E}[\mathrm{lp}_k] &= \frac{\alpha\Delta}{\alpha-\beta}\left[(1-\beta)^k - (1-\alpha)^k\right]. \end{aligned} \tag{25}$$

For every finite $k \geq 1$, the expected learning-progress state has the sign of Δ, and it converges to zero after the slow mastery state catches up to the new level.

Proof. Taking expectations in the mastery recursion and solving the first-order difference equation gives the first line of (25). Thus the expected residual at visit j is

$$\mathbb{E}[d_j] = \mu_1 - \mathbb{E}[\mathrm{hm}_{j-1}] = \Delta(1-\beta)^{j-1}. \tag{26}$$

Unrolling the progress recursion then gives

$$\begin{aligned} \mathbb{E}[\mathrm{lp}_k] &= \alpha \sum_{j=1}^{k} (1-\alpha)^{k-j}\mathbb{E}[d_j] \\ &= \frac{\alpha\Delta}{\alpha-\beta}\left[(1-\beta)^k - (1-\alpha)^k\right]. \end{aligned} \tag{27}$$

Because $0 < 1-\alpha < 1-\beta < 1$, the bracket is positive for every finite $k \geq 1$ and tends to zero, proving the sign and limit claims.

Remark. Proposition 2 formalizes LP as a transient change signal rather than a competence score: an upward shift produces a positive pulse, a downward shift a negative pulse, and a stable new level eventually returns to zero. It is an expectation-level response under an idealized persistent shift, not a claim that every noisy observed trajectory is monotone or that the shift was caused by the optimizer.

Assumption 1 (Local stationarity)**.** For successive visits to one prompt, let

$$p_k = \frac{1}{G}\sum_{i=1}^{G} r_{k,i}, \qquad r_{k,i} \overset{\text{i.i.d.}}{\sim} \mathrm{Bernoulli}(\mu), \tag{28}$$

with outcomes independent across visits. Hence $\mathbb{E}[p_k] = \mu$ and $\mathrm{Var}[p_k] = s^2 = \mu(1-\mu)/G$.

Proposition 3 (Exact stationary noise floor)**.** *Under Assumption 1 and $0 < \alpha, \beta < 1$, the two-timescale state is mean-stable and has stationary moments*

$$\begin{aligned} \mathbb{E}[\mathrm{hm}_k] \to \mu, \quad \mathrm{Var}[\mathrm{hm}_\infty] &= \frac{\beta}{2-\beta}\, s^2, \\ \mathbb{E}[\mathrm{lp}_k] \to 0, \qquad \mathrm{Var}[\mathrm{lp}_\infty] &= \frac{2\alpha^2}{(2-\alpha)(2-\beta)(\alpha+\beta-\alpha\beta)}\, s^2. \end{aligned} \tag{29}$$

Proof. Let $\xi_k = p_k - \mu$ and $z_k = (h_k, l_k)^\top = (\mathrm{hm}_k - \mu, \mathrm{lp}_k)^\top$. The centered state obeys

$$z_k = \underbrace{\begin{pmatrix} 1-\beta & 0 \\ -\alpha & 1-\alpha \end{pmatrix}}_{M} z_{k-1} + \underbrace{\begin{pmatrix} \beta \\ \alpha \end{pmatrix}}_{b} \xi_k. \tag{30}$$

The eigenvalues of M are $1-\beta$ and $1-\alpha$, both inside the unit circle, so $\mathbb{E}[z_k] \to 0$. Independence of ξ_k and z_{k-1} implies that the unique stationary covariance $\Sigma = \mathrm{Cov}(z_\infty)$ solves

$$\Sigma = M\Sigma M^\top + s^2 bb^\top. \tag{31}$$

Writing $V_h = \Sigma_{11}$, $C = \Sigma_{12}$, and $V_l = \Sigma_{22}$, Eq. (31) is equivalent to

$$\begin{aligned} V_h &= (1-\beta)^2 V_h + \beta^2 s^2, \\ C &= (1-\beta)(1-\alpha)C - \alpha(1-\beta)V_h + \alpha\beta s^2, \\ V_l &= (1-\alpha)^2 V_l + \alpha^2(V_h + s^2) - 2\alpha(1-\alpha)C. \end{aligned} \tag{32}$$

Solving these three scalar equations in order yields

$$\begin{aligned} V_h &= \frac{\beta}{2-\beta} s^2, \\ C &= \frac{\alpha\beta}{(2-\beta)(\alpha+\beta-\alpha\beta)} s^2, \\ V_l &= \frac{2\alpha^2}{(2-\alpha)(2-\beta)(\alpha+\beta-\alpha\beta)} s^2, \end{aligned} \tag{33}$$

which proves (29).

Remark. For the reported $(\alpha, \beta) = (0.4, 0.2)$, the exact variances are $\mathrm{Var}[\mathrm{hm}_\infty] = s^2/9$ and $\mathrm{Var}[\mathrm{lp}_\infty] = 25s^2/117$. As $\beta \to 0$, the latter reduces to $\alpha s^2/(2-\alpha)$, the constant-baseline limit. Thus LP is centered at zero when competence is locally stationary but retains a finite rollout-induced noise floor. A sustained same-sign trajectory is evidence against the stationary model only after accounting for this noise; an isolated positive or negative value is not proof of learning or forgetting.

Signal-potential sampling.

Proposition 4 (Signal potential ranks informative groups)**.** *Let the $G \geq 2$ rollout rewards for a prompt be i.i.d. Bernoulli(p), and define $g(p) = 2\sqrt{p(1-p)}$. Then*

$$\mathrm{Var}[r] = p(1-p) = \big(g(p)/2\big)^2, \qquad \pi_G(p) = 1 - p^G - (1-p)^G, \tag{34}$$

where $\pi_G(p)$ is the probability that a group contains both a success and a failure. Both $g(p)$ and $\pi_G(p)$ are symmetric about $p = 1/2$, increase as p approaches $1/2$, and attain their maxima there. They therefore induce the same weak ranking of prompts for fixed G.

Proof. The variance identity is standard. A binary group is degenerate exactly when all G outcomes are successes or all are failures, with probability $p^G + (1-p)^G$; taking its complement gives $\pi_G(p)$. Symmetry follows by replacing p with $1-p$. On $0 \leq p < 1/2$, $p(1-p)$ is increasing and

$$\pi_G'(p) = G\big[(1-p)^{G-1} - p^{G-1}\big] > 0, \tag{35}$$

so $\pi_G(p)$ is also increasing. Symmetry gives the corresponding decrease on $1/2 < p \leq 1$. Thus both quantities rank prompts solely by proximity to one half.

Remark. For the usual centered group-reward advantage, a binary group has nonzero group-relative advantages if and only if it is non-degenerate. Historical mastery is only a plug-in estimate of the current p, so prioritizing high $g(\mathrm{hm})$ increases the estimated chance of such a group but does not guarantee a nonzero realized parameter gradient. The factor of two cancels under proportional sampling normalization.

Advantage reweighting.

Proposition 5 (Bounded, sign-preserving reweighting)**.** *Assume* $0 < \epsilon_p < 1/2$, $\gamma \geq 0$, $\kappa \geq 0$, *and* $0 \leq \beta_p \leq \alpha_p$. *Then*

$$\begin{aligned} \epsilon_p^\gamma \leq w_{\text{diff}}^x &\leq (1-\epsilon_p)^\gamma, \\ 1 \leq w_{\text{prog},t}^x &\leq 1+\alpha_p, \\ 0 < \epsilon_p^\gamma \leq w_t^x &\leq (1-\epsilon_p)^\gamma(1+\alpha_p) \leq 1+\alpha_p. \end{aligned} \tag{36}$$

Consequently, for every response,

$$\operatorname{sign}\big(\widetilde{A}_{i,t}^x\big) = \operatorname{sign}\big(A_{i,t}^{x,\text{GRPO}}\big), \qquad |\widetilde{A}_{i,t}^x| \leq (1+\alpha_p)|A_{i,t}^{x,\text{GRPO}}|. \tag{37}$$

Proof. Clipping gives $1-\bar{p}_0^x \in [\epsilon_p, 1-\epsilon_p]$; raising this positive base to $\gamma \geq 0$ gives the first line of (36). In either progress branch, the argument of $\tanh$ is nonnegative, so its value lies in $[0,1)$. The improving branch is therefore bounded by $1+\alpha_p$, while the regressing branch is bounded by $1+\beta_p \leq 1+\alpha_p$. Multiplying the two positive factors gives the final line. Since $\widetilde{A}_{i,t}^x = w_t^x A_{i,t}^{x,\text{GRPO}}$ with $w_t^x > 0$, the sign is unchanged and the magnitude bound follows.

Remark. Under the reported $(\epsilon_p, \gamma, \alpha_p) = (10^{-3}, 0.5, 1.0)$, the deterministic bound is approximately $0.0316 \leq w_t^x \leq 1.9990$. LP-Reweighting can therefore attenuate an initially easy prompt or amplify an improving hard prompt, but it neither reverses an advantage nor creates a nonzero advantage from a zero one.

Proposition 6 (Exact prompt-reweighted objective)**.** *Fix* $\mathcal{B}_t$ *and its sampled rollouts, and treat* $A_{i,t}^{x,\text{GRPO}}$ *and* w_t^x *as detached constants during the surrogate-policy update. Let* $\rho_{i,j}^x = \rho_{i,j}^x(\theta)$ *be the token-level importance ratio, and define prompt* x*'s unnormalized clipped GRPO term by*

$$\mathcal{J}_x^{\text{GRPO}}(\theta) = \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_j \min\Big(\rho_{i,j}^x A_{i,t}^{x,\text{GRPO}}, \operatorname{clip}(\rho_{i,j}^x, 1-\epsilon_c, 1+\epsilon_c) A_{i,t}^{x,\text{GRPO}}\Big). \tag{38}$$

Then the LP-GRPO objective is exactly

$$\mathcal{J}_{\text{LP-GRPO}}(\theta; \mathcal{B}_t) = \frac{1}{|\mathcal{B}_t| G} \sum_{x \in \mathcal{B}_t} w_t^x \mathcal{J}_x^{\text{GRPO}}(\theta). \tag{39}$$

Where differentiable, its gradient is the same weighted sum of the per-prompt GRPO gradients.

Proof. For each token, let $a = \rho_{i,j}^x A_{i,t}^{x,\text{GRPO}}$ and let b be its clipped counterpart. Proposition 5 gives $w_t^x > 0$, and positive scaling commutes with the minimum:

$$\min(w_t^x a, w_t^x b) = w_t^x \min(a, b). \tag{40}$$

The same w_t^x applies to every response and token of prompt x, so it factors out of that prompt's sums and gives (39). Because the multiplier is detached, differentiating the exact equality yields the stated gradient relation.

Remark. LP-Reweighting is a per-prompt objective weight on an otherwise unchanged clipped GRPO surrogate. It changes relative prompt contributions, not the within-prompt clipping branch, and it is not a claim that Adam uses a different learning rate for each prompt. Together with LP-Sampling, the effective prompt emphasis is determined by both selection frequency and this objective weight.

Propositions 1–3 establish that the LP state is bounded, directionally responsive, and stable with a quantifiable rollout-noise floor. Proposition 4 justifies the sampler's signal potential as a plug-in ranking of non-degenerate groups, while Propositions 5 and 6 establish safe and exact prompt-level objective reweighting. None of these local properties alone implies global policy convergence or causal performance improvement.

D Detailed Allocation and Revisit Diagnostics

D.1 Post-Hoc State Definitions

For each retained prompt group, let p_0 be its offline pass rate, p_t its pass rate at the current visit, and $\Delta_t = p_t - p_0$. We assign mutually exclusive analysis labels by applying the following conditions in order: *mastered* if $p_0 > 0.8$ and $\Delta_t \geq 0$; *regressing* if $\Delta_t < -0.05$; *breakthrough* if $p_0 < 0.2$ and $\Delta_t > 0.15$; *progress* if $\Delta_t > 0.05$; and *plateau* otherwise. Breakthrough and progress groups are collectively called *improving states*. Because both pass rates are computed from eight rollouts, the positive and negative progress thresholds correspond to changes of at least one correct response, while the breakthrough threshold requires an increase of at least two. These labels are used only for post-hoc analysis; they are neither sampler inputs nor exact indicators of reward variance or gradient magnitude.

D.2 Allocation and Training Stability

Table 7 summarizes reward and improving-state allocation, while Table 8 and Figure 9 report the corresponding on-policy stability diagnostics under the same update-360 cutoff.

Method	Reward (F/L)	Improving (k)	Last-50
GRPO	.473/.542	18.68	42.47
DAPO	.441/.519	17.58	40.63
CDAS	.585/.386	23.74	40.31
LP-GRPO	.594/.684	**23.90**	**55.69**

Table 7: Reward and improving-state allocation through update 360. F/L denotes the first/last 50 updates, Improving is the cumulative number of improving-state groups in thousands, and Last-50 is their percentage over the final 50 updates.

Method	Entropy (F/L)	Length (F/L)
GRPO	.334/.325	2041/2598
DAPO	.357/.323	1198/1302
CDAS	.405/.408	2437/4245
LP-GRPO	.467/.571	2337/2751

Table 8: Training-stability diagnostics through update 360. F/L denotes the first/last 50 updates. Entropy is the mean policy entropy, and Length is the mean response length in tokens. Only LP-GRPO increases entropy over training while keeping response length nearly flat.

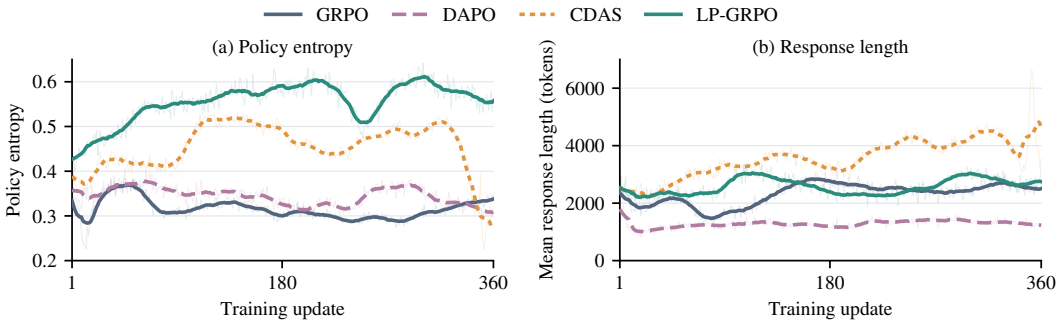


Figure 9: Training dynamics through update 360. (a) Mean policy entropy and (b) mean response length. Bold curves are centered 17-update moving averages of the faint traces.

D.3 Stagewise Revisit Diagnostics

Table 9 follows the mechanism from pool coverage to optimizer use. Coverage expands throughout training while the fresh-prompt share decreases, selected prompts retain higher sampling scores than the eligible pool, and by update 360 most groups reuse progress stored at an earlier visit for LP-Reweighting.

Update	Coverage	Zero cov.	Fresh	Visits	Sel. score	Elig. score	Stored LP	Mean w
40	16.80	8.28	96.09	1.01	0.732	0.636	5	1.091
120	42.33	23.58	66.41	1.20	0.828	0.652	43	1.096
240	65.30	42.42	28.13	1.55	0.870	0.671	92	1.154
360	77.66	56.03	20.31	1.96	0.853	0.681	102	1.067

Table 9: Stagewise LP-Sampling and LP-Reweighting diagnostics through update 360. Coverage, zero-prior coverage, and fresh-prompt share are percentages; Visits is the mean visit count over seen prompts; Sel./Elig. score are the mean sampling scores of the selected batch and eligible pool; and Stored LP counts groups, out of 128, whose advantage multiplier uses progress from an earlier visit.

E Hyperparameter Checks

On MathVision, with the 360-step budget held fixed, the full setting reaches 61.35. Setting $\lambda = 0$ removes the learning-progress boost and reaches 57.30. With $\alpha_p + \beta_p = 1.4$ fixed, the symmetric factors $(0.7, 0.7)$ reach 58.90 and the reversed factors $(0.4, 1.0)$ reach 57.70. Replacing $(\alpha, \beta) = (0.4, 0.2)$ with $(0.2, 0.4)$ or $(0.3, 0.3)$ reaches 59.21 and 59.70.

Parameter	Setting	Accuracy (%)
Full setting		**61.35**
λ	0	57.30
(α_p, β_p)	$(0.4, 1.0)$	57.70
	$(0.7, 0.7)$	58.90
(α, β)	$(0.2, 0.4)$	59.21
	$(0.3, 0.3)$	59.70

Table 10: MathVision accuracy at step 360. The full setting uses $\lambda = 1$, $(\alpha_p, \beta_p) = (1.0, 0.4)$, and $(\alpha, \beta) = (0.4, 0.2)$. Each block changes only the indicated group.

F Allocation Diagnostics

For each retained group, let p_0 be its offline pass rate, p_t its pass rate at the visit, and $\Delta_t = p_t - p_0$. The post-hoc labels, applied in order, are mastered if $p_0 > 0.8$ and $\Delta_t \geq 0$; regressing if $\Delta_t < -0.05$; breakthrough if $p_0 < 0.2$ and $\Delta_t > 0.15$; progress if $\Delta_t > 0.05$; and plateau otherwise. Breakthrough and progress are the improving states. These labels are not inputs to LP-GRPO.

Relative to GRPO, LP-GRPO raises the progress-state share from 31.30% to 42.34% and lowers the plateau share from 31.24% to 17.51%. The mastered and breakthrough shares stay close to GRPO. Regressing states are 14.89% of LP-GRPO's groups.

G Reproducibility Scope

The artifact contains the paper sources, figures, implementation snapshots, resolved configuration, and analysis scripts. External model weights, datasets, and benchmark packages are referenced and not redistributed. Table 6 records the hardware and seeds. Each trained checkpoint is from one run.

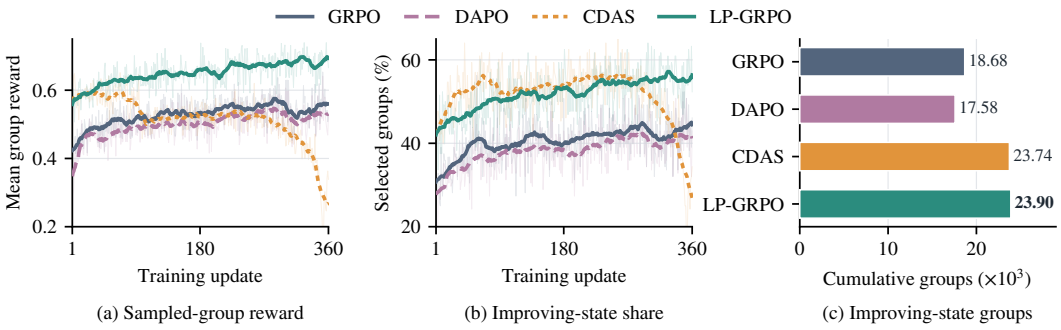


Figure 10: Allocation through step 360. (a) Mean reward of sampled groups. (b) Improving-state share. (c) Cumulative improving-state groups. Bold curves are centered averages of the faint traces.

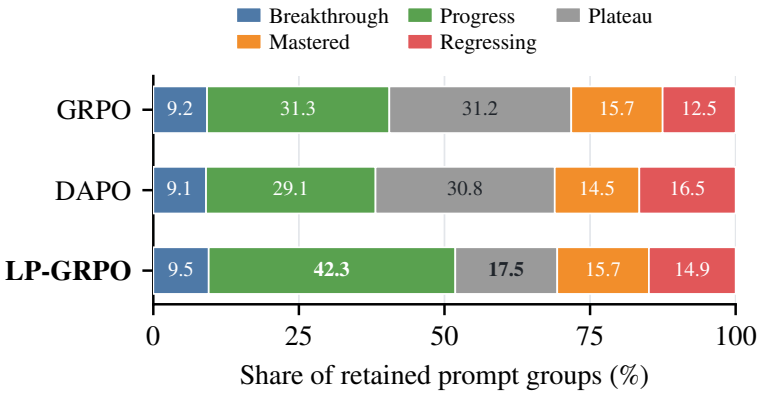


Figure 11: Composition of the matched 46,080-group budget. Each bar partitions retained groups into five post-hoc states.

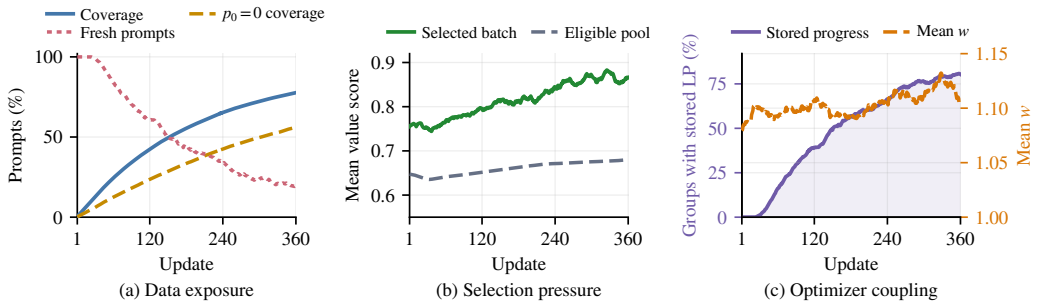
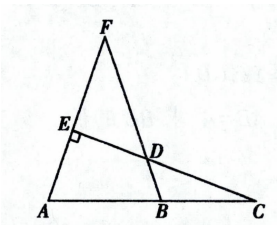


Figure 12: Shared prompt state through step 360. (a) Pool coverage and the fresh-prompt share. (b) Mean sampling score of selected prompts against the eligible pool. (c) Groups whose multiplier reuses stored progress, and the mean multiplier w.

H Qualitative Output Evolution Across Revisits

We show three real prompts whose group accuracies improve across revisits; each accuracy counts correct responses among the eight original rollouts. Excerpts are verbatim apart from whitespace and mathematical typesetting, with ellipses marking omissions.

H.1 Geometry



Question. Point C is on the extension of AB. Line CE is perpendicular to AF at E and intersects BF at D. If $\angle F = 40^\circ$ and $\angle C = 20^\circ$, find $\angle FBC$.

Ground truth. 110°.

Training visits. $0/8 \to 2/8 \to 4/8 \to 6/8$.

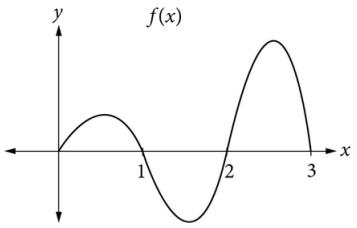
Early output (incorrect).

"We only need to find $\angle FBC$, which, as we said, lies in triangle FBC, with angles at F and C known. So: $\angle FBC = 180^\circ - 40^\circ - 20^\circ = 120^\circ$. ... Thus, the answer is 120°. ... $\boxed{120^\circ}$ ×"

Late output (correct).

"Point E is on AF, and $CE \perp AF$, so $\angle AEC = 90^\circ$. So in triangle AEC, $\angle AEC = 90^\circ$, $\angle C = 20^\circ$, so $\angle A = 180-90-20 = 70^\circ$. ... In triangle ABF, $\angle A = 70^\circ$, $\angle F = 40^\circ$, so $\angle ABF = 180-70-40 = 70^\circ$. Therefore, $\angle FBC = 180^\circ - 70^\circ = 110^\circ$. ... $\boxed{110^\circ}$ ✓"

H.2 Graph Interpretation



Question. Which statement about the plotted function is true? (A) $f''(x) > 0$ for $1 < x < 2$; (B) $f''(x) < 0$ for $1 < x < 2$; (C) $f'(x) > 0$ for $0 < x < 1$ and $2 < x < 3$; (D) $f'(x) < 0$ for $0 < x < 1$ and $2 < x < 3$.

Ground truth. A.

Training visits. $0/8 \to 1/8 \to 6/8$.

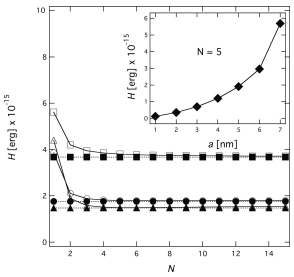
Early output (incorrect).

"From $x = 1$ to $x = 2$: the graph is decreasing ... The graph is bending downward—meaning the slope is becoming more negative, so concave down $\to f''(x) < 0$. ... So only option B seems correct. ... $\boxed{B}$ ×"

Late output (correct).

"The function is decreasing, but its slope is becoming less negative (i.e., slope is increasing from negative to zero at the minimum, then becoming positive). So the first derivative is increasing $\to f''(x) > 0$. ... From $x = 1$ to $x = 2$, the curve bends upward—similar to a U shape, so concave up everywhere between 1 and 2. ... Thus A is true. ... $\boxed{A}$ ✓"

H.3 SCIENTIFIC-PLOT READING



Question. For particles of size $a = 2\,\mathrm{nm}$, how does the energy H change as the number of particles N increases? (A) H increases rapidly; (B) H decreases; (C) H remains constant; (D) H initially decreases and then increases.

Ground truth. C.

Training visits. $0/8 \to 1/8 \to 3/8 \to 6/8$.

Early output (incorrect).

"Looking at the legend or the plot, the data points marked with triangles (▲) seem to correspond to $a = 2$ nm ... Following the triangle-marked curve on the main graph: At $N = 2$, H is roughly 2.2; at $N = 4$, it's ~ 1.8; at $N = 6$, ~ 1.7; and then it flattens out around $N = 8$. ... The best match among the options is B. ... $\boxed{B}$ ×"

Late output (correct).

"Looking at the data points marked with circles ..., we see that for N from 2 up to 14, the H value ... stays roughly constant around 2×10^{-15} erg. ... So, for $a = 2$ nm, as N increases, H does not change—it stabilizes. ... Therefore, the correct answer is C. ... $\boxed{C}$ ✓"