Learning to Solve, Forgetting to Retain: Correct-Set Turnover in RLVR

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) improves the ability of large language model, yet headline accuracy gains often conceal a hidden cost: previously solved problems quietly become unsolvable as training proceeds. We frame this phenomenon as *correct-set turnover*, representing the coupled dynamics of solution acquisition and regression over the mastered set. Under this view, retention becomes an explicit optimization target alongside acquisition. We analytically and empirically establish the *repair-window principle*: the cost of restoring a regressed prompt grows sharply with review delay, defining a low-cost window that standard RLVR pipelines fail to exploit. To address this, we propose **ReMind**, a retention-aware review mechanism that tracks mastered prompts and periodically reintroduces them to **remind** the model of previous solutions. By utilizing pre-rollout batch replacement, ReMind incurs zero additional rollout overhead. Evaluated across 20 benchmarks spanning image-text, video, and text-only tasks with Qwen3-VL and Qwen2.5-Math, ReMind consistently improves performance over GRPO, DAPO, and replay baselines, demonstrating robust generalizability across modalities and algorithms.

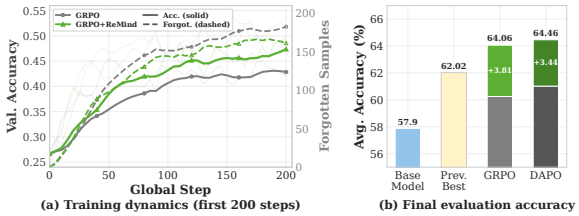


Figure 1: (a) Training dynamics during the first 200 steps: validation accuracy (solid lines, left axis) and correct-set loss (dashed lines, right axis). GRPO accumulates forgetting steadily while accuracy plateaus; ReMind reduces forgetting and sustains accuracy gains under the same rollout budget. (b) Average final accuracy across eight multimodal reasoning benchmarks. ReMind improves both GRPO and DAPO baselines.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a standard paradigm for improving the reasoning capabilities of large language models (Guo et al., 2025; Shao et al., 2024; Yu et al., 2025). Headline accuracy gains, however, conceal a less visible cost: previously solvable problems can become unsolvable as training proceeds. RLVR has been shown to narrow the solution distribution, raising Pass@1 while reducing Pass@K (Yue et al., 2025b; Cui et al., 2025), and acquiring new reasoning skills can interfere with previously stable ones (Hu et al., 2026). Forgetting in RLVR is thus a systematic, intra-task consequence of policy updates that concentrate probability mass on newly reinforced solutions, distinct from the cross-task setting of continual learning.

The forgetting view alone treats this as a side effect to be minimized. We find it more useful to frame the underlying phenomenon symmetrically: at each step the model's accuracy is the net of two ongoing flows, an acquisition flow (problems newly solved) and a regression flow (problems no longer solved). We refer to this joint dynamics over the model's correct set as *correct-set turnover*, of which forgetting is the regression side. This lets us pose retention as an optimization problem over two coupled flows: minimize regression while preserving acquisition, by identifying which mastered prompts are at risk and *when* to intervene.

Human cognition offers a useful starting point for thinking about *when*. Spaced-repetition research has long shown that reviewing material before it is fully forgotten is far cheaper than relearning it, with review cost growing sharply with the delay before intervention (Ebbinghaus, 1885; Cepeda et al., 2006). We hypothesize that an analogous asymmetry holds for mastered prompts under RLVR. Parameter updates driven by other samples

cause a mastered prompt's success probability to drift downward through gradient interference (formalized in §3.2), so an early review can restore the prompt cheaply, whereas a review after full regression requires effort comparable to first-time learning. Our controlled experiment in Figure 2 confirms this asymmetry in RLVR training. The design principle that follows is clear: a lightweight review mechanism that reaches mastered prompts before they leave the low-cost repair window can maintain retention without measurably slowing acquisition. Standard RLVR pipelines (Shao et al., 2024) fall short here, revisiting prompts only passively across epochs with no record of mastery.

We propose **ReMind**, a retention-aware review mechanism that periodically reviews mastered samples to **remind** the model of previously learned solutions. ReMind operationalizes this principle by treating the mastered set as an explicit, trackable state. It maintains a review memory of fully mastered prompts and, on designated review steps, replaces a small fraction of the main batch with prompts drawn from this memory prior to rollout generation. Because review samples replace rather than augment the batch, ReMind incurs zero additional rollout cost over standard training. Finally, a self-correcting update rule keeps the memory dynamically focused on unstable prompts: samples that remain fully correct are released, while regressed ones are retained for future re-checks.

We evaluate ReMind across a comprehensive suite of 20 benchmarks spanning image-text, video, and text-only mathematical reasoning, utilizing both Qwen3-VL-8B-Instruct and Qwen2.5-7B-Math. As shown in Figure 1(a), correct-set loss accumulates steadily during standard GRPO training while accuracy plateaus, whereas ReMind reduces forgetting and sustains accuracy gains throughout. On final evaluation (Figure 1(b)), ReMind consistently improves over GRPO, DAPO, and replay baselines, with gains attributable primarily to reduced forgetting rather than faster acquisition. The improvement holds under both GRPO and DAPO, confirming that retention-aware review and online filtering are complementary.

Our contributions are as follows.

- We formalize *correct-set turnover* in RLVR as the joint dynamics of acquisition and regression. Using a zero-overhead, graded mastery metric, we establish the *repair-window principle*: the cost of repairing regressed prompts grows sharply with delayed intervention.
- We propose ReMind, a retention-aware review mechanism that maintains a mastery-keyed review memory and uses pre-rollout replacement to detect and repair regression without increasing the rollout budget.
- ReMind improves late-stage accuracy and retention over standard RLVR baselines and replay methods across 20 benchmarks, demonstrating robust generalizability across different modalities, models, and algorithmic backbones.

2 Related Work

RL for LLMs. RLVR has become a standard post-training method for reasoning models (Shao et al., 2024; Guo et al., 2025), with GRPO replacing the learned value model by group-relative comparison. Recent variants improve update stability and informativeness through dynamic sampling, clip-higher, and length-aware shaping (DAPO (Yu et al., 2025)), normalization bias correction (Liu et al., 2025), and diagnostics on entropy and token-level objectives (Cui et al., 2025; Dong et al., 2026). A related filtering and curriculum line (Yu et al., 2025; Bae et al., 2025; Jiang et al., 2025) selects prompts whose rollouts are neither all correct nor all incorrect to keep useful gradients in the batch, treating uniformly correct prompts as low-value for the *next update*. Under the correct-set turnover view, those same prompts are exactly the ones whose *retention* must be checked: a prompt leaves the batch because it is solved, and later leaves the correct set because it has regressed. ReMind targets this overlooked side of the same selection decision.

Replay and forgetting. Experience replay reuses past data for efficiency (Lin, 1992; Mnih et al., 2015), with prioritization based on estimated learning value (Schaul et al., 2015). Recent LLM RL methods replay successful or informative trajectories along various axes, including verified solutions (Zhang et al., 2025a), difficulty and entropy buckets (Zhan et al., 2025), response reuse and early-state replay (Zhang et al., 2025b; Dou et al., 2025), and age-decayed priorities (Ma et al., 2026), but none defines priority by whether a once-mastered prompt has regressed. This differs from continual learning, where forgetting arises from sequential exposure to disjoint tasks (Kirkpatrick et al., 2017; Rebuffi et al., 2017); we study regression within a single RL training distribution. Our

framing is closest to example-forgetting analysis, where forgetting is a sample-level 1→0 transition under single-evaluation labels (Toneva et al., 2019). This binary view is natural in supervised settings, but in group-based RLVR each prompt is already evaluated through multiple rollouts; treating these as a graded mastery estimate exposes partial regression that single-evaluation definitions cannot register, which ReMind exploits to act while repair is still cheap. Recent RLVR work also reports broader regression phenomena, including cross-task capability loss (Phan et al., 2025), narrowed pass@k distributions (Yue et al., 2025b), and collapse dynamics (Hu et al., 2026), but does not turn sample-level mastery into an online training state for retention decisions.

3 Correct-Set Turnover and ReMind

3.1 Definition and Observation

Let $\mathcal{P}$ denote the training set. For each sample $p \in \mathcal{P}$, we write $q_t(p) = \Pr[\pi_{\theta_t} \text{ solves } p]$ for the success probability under the current policy π_{θ_t}, estimated with K rollouts as $\hat{q}_t(p) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}[r_i = 1]$. A sample is *mastered* at step t if all rollouts are correct ($\hat{q}_t(p) = 1$), yielding the mastered set $M_t = \{p \in \mathcal{P} : \hat{q}_t(p) = 1\}$. Two properties of $\hat{q}_t(p)$ matter for what follows. It is *graded*: a sample at $\hat{q}_t = 12/16$ is already drifting from mastery but would still appear "correct" under a single evaluation with probability 0.75, whereas the binary signal used in prior example-forgetting work (Toneva et al., 2019) collapses this regime. It is also *free*: $\hat{q}_t(p)$ is the same per-prompt success rate GRPO computes for advantage estimation, so retention can be tracked as an operational state without any auxiliary evaluation.

Comparing M_t with a later $M_{t+\Delta}$ decomposes the net accuracy change into two opposing flows: *acquisition*, where samples enter the mastered set ($p \notin M_t,\ p \in M_{t+\Delta}$), and *regression*, where samples leave it ($p \in M_t,\ p \notin M_{t+\Delta}$). We call this continual flux *correct-set turnover* and measure it with the retention rate:

$$\text{Retention}_t = \frac{|M_t \cap M_{t+\Delta}|}{|M_t|},$$
$$\text{Regression}_t = 1 - \text{Retention}_t. \qquad (1)$$

Figure 1(a) provides direct evidence. Training Qwen3-VL-8B-Instruct with standard GRPO, we track validation accuracy alongside the count of samples that were once solved but are currently wrong (correct-set loss). The two signals reveal a telling asymmetry: accuracy rises and then plateaus after roughly 100 steps, yet correct-set loss climbs steadily throughout the same window, indicating that new acquisitions are increasingly offset by regressions on previously mastered samples.

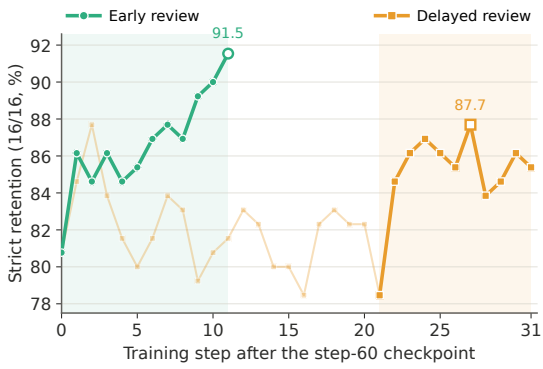


Figure 2: Strict retention (16/16 rollouts correct) of a fixed set of mastered samples under early and delayed review. Early review includes these samples from step 0; delayed review withholds them for 20 steps (shaded region). The retention gap during the delay period confirms that timely review prevents deeper regression.

3.2 The Repair-Window Principle

Once a sample p is mastered at step t_0 and subsequently absent from the training batch, its success probability $q_t(p)$ remains sensitive to parameter updates driven by other training samples. A first-order Taylor expansion yields:

$$q_{t+1}(p) \approx q_t(p) - \alpha \nabla_\theta q_t(p)^\top g_t, \qquad (2)$$

where g_t is the batch gradient and α the learning rate. The inner product $\nabla_\theta q_t(p)^\top g_t$ captures *gradient interference*: updates that improve other samples may shift parameters away from configurations that produce correct solutions for p. When p receives no reinforcing signal, $q_t(p)$ drifts downward as a continuous process, a drift directly trackable through the graded measure of §3.1.

This drift implies that the cost of repair grows with the delay. Consider a sample mastered at t_0 with $q_{t_0}(p) \approx 1$. After k unreviewed steps, the success probability has drifted to approximately $1 - k\alpha\delta(p)$, where $\delta(p) > 0$ is an unknown but positive scalar summarizing the per-step interference magnitude for sample p. If review occurs while q is still high (small k), the model's own rollouts remain predominantly correct, providing a self-reinforcing gradient that restores q within one or two updates. Because all rollouts are likely

correct in this regime, the within-group advantage variance is near zero, so the review sample adds almost no gradient noise to the rest of the batch. If review is delayed until q has fallen well below the mastery threshold, the model can no longer produce correct rollouts reliably. Relearning then requires effort comparable to first-time learning. While a precise convergence rate depends on optimization geometry and the RLVR reward landscape, a standard estimate places this at the order of $O(1/\alpha)$ steps, which combined with the early-repair cost $O(k\alpha\delta)$ above gives a heuristic ratio of $O(k\alpha^2\delta)$. Under typical hyperparameters (e.g., $\alpha \sim 10^{-6}$), this ratio is small for any reasonable k and δ, suggesting that **early intervention is substantially cheaper than late remediation**.

We verify this prediction with a controlled experiment. From a step-60 GRPO checkpoint, we select samples that are stably mastered ($16/16$ rollouts correct). We then compare two conditions: *early review*, where these samples are included in training immediately, and *delayed review*, where the model trains on background data for 20 steps before they are reintroduced. The delayed condition resembles the natural schedule in multi-epoch RL training. Figure 2 shows strict retention over 30 subsequent training steps. Early review maintains retention between 85% and 91% throughout, whereas the delayed condition drops to 78–84% during the silent period. After delayed review begins at step 20, retention recovers but does not fully close the gap: at maximum difference, 14 fewer samples are retained compared to early review.

This yields a clear design principle: a regular review mechanism that reaches mastered samples before they leave the repair window can maintain high retention at low cost. The remainder of this section instantiates this principle as ReMind.

3.3 ReMind

ReMind adds a single mechanism to standard GRPO training: a review queue that periodically reintroduces previously mastered samples into the training batch before rollout generation. Reviewed samples pass through the same rollout, reward, and policy-update pipeline as regular training samples, so ReMind operates within the same per-step rollout budget as the baseline. Figure 3 illustrates the pipeline. Crucially, ReMind introduces no new measurement: both the mastery criterion $\hat{q}_t(p) = 1$ and the regression criterion $\hat{q}_t(p) < 1$ are read directly from the group rollouts GRPO already produces, so the same multi-rollout structure that drives GRPO's gradient signal also supplies the graded resolution needed to act within the low-cost repair window.

The method has three components: a FIFO review queue that stores mastered samples (§3.3.1), a batch-construction rule that mixes review and fresh samples on designated review steps (§3.3.2), and an update rule that routes regressed samples back into the queue for future repair (§3.3.3).

3.3.1 Review Queue

Following the mastery definition in §3.1, when a sample from the standard training stream satisfies $\hat{q}_t(p) = 1$, ReMind appends it to the review queue with probability p_{add}. The queue stores sample identifiers only; each subsequent review generates fresh rollouts from the current policy rather than replaying stored trajectories. FIFO ordering ensures that earlier-mastered samples are reviewed first, bounding the number of intervening policy updates before each sample is checked, which aligns with the repair-window analysis: the queue naturally schedules review while $q_t(p)$ remains high and repair cost is low.

3.3.2 Batch Construction

Every f steps is designated a *review step*. On non-review steps, the batch is drawn entirely from the training sampler. On a review step, ReMind allocates a fraction r of the batch to review samples drawn from the queue Q_t, forming a mixed batch:

$$\widetilde{B}_t = B_t^{\text{fresh}} \cup B_t^{\text{review}},$$
$$|B_t^{\text{review}}| = \min\big(\lfloor r \cdot |\widetilde{B}_t| \rfloor,\ |Q_t|\big), \qquad (3)$$

where B_t^{fresh} contains samples from the standard stream and B_t^{review} contains samples dequeued from the head of Q_t. This mixed batch then proceeds through rollout, reward computation, and policy update without modification. The review ratio r and period f together control the fraction of total training devoted to retention review.

3.3.3 Queue Update

After reward computation, the queue is updated according to three cases (lower panel of Figure 3). (1) Any sample in $\widetilde{B}_t$ that is *newly mastered* ($\hat{q}_t(p) = 1$) and not already in the queue is enqueued with probability p_{add}. (2) A review sample from B_t^{review} that *retains* full accuracy is removed from the queue, as its retention has been confirmed. (3) A review sample that *regresses* ($\hat{q}_t(p) < 1$) is

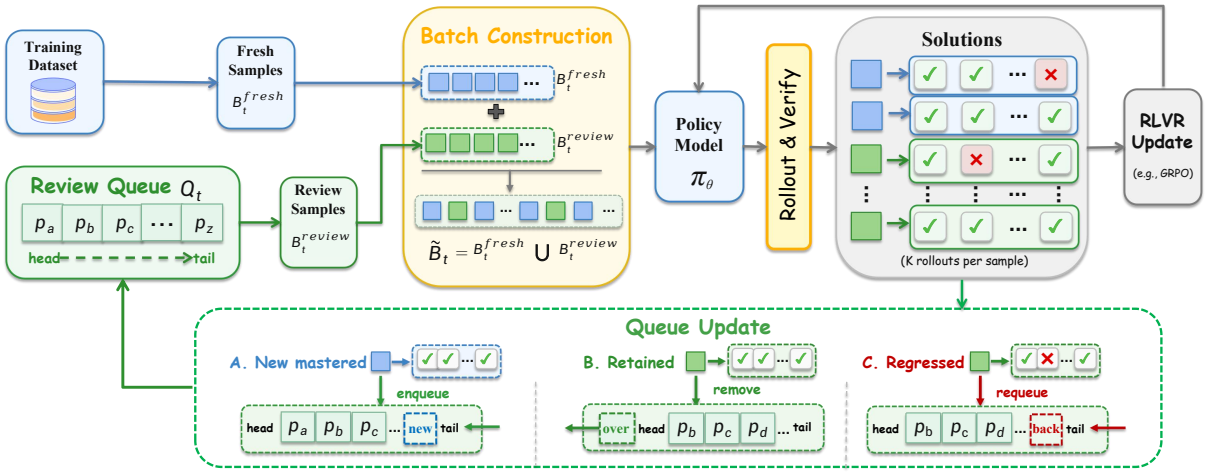


Figure 3: Overview of ReMind. The upper half shows the main training loop: fresh samples and review samples from the queue are combined into a mixed batch $\widetilde{B}_t$, passed through the policy model for rollout generation, verified, and used for the RLVR policy update. The lower half shows the three queue-update cases after verification: (A) newly mastered samples are enqueued, (B) retained samples are removed from the queue, and (C) regressed samples are reinserted at the tail for future review.

reinserted at the queue tail for future repair. This ensures regressed samples cycle back for additional review while stably retained samples exit.

4 Experiments

4.1 Experimental Setups

Datasets. We train on MMFineReason-123K (Lin et al., 2026), a difficulty-filtered subset of MMFineReason-1.8M retains only samples unsolvable by Qwen3-VL-4B-Thinking (Bai et al., 2025) across four rollouts.We evaluate on eight image-text reasoning benchmarks covering math reasoning (MathVista (Lu et al.), MathVision (Wang et al., 2024), WeMath (Qiao et al., 2025), MathVerse (Zhang et al., 2024)) and general multimodal understanding (MMMU-Pro (Yue et al., 2025a), MMBench (Liu et al., 2024), MM-Star (Chen et al., 2024), ZeroBench (Roberts et al., 2025)).

Models and baselines. All experiments use Qwen3-VL-8B-Instruct (Bai et al., 2025) as the base policy, with every method implemented on a shared GRPO-style backbone. We compare against two on-policy baselines, GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) (which adds dynamic sampling and online filtering), and three replay methods that reuse historical trajectories: ExGRPO (Zhan et al., 2025), RLEP (Zhang et al., 2025a), and RePO (Li et al., 2025).

Implementation. We build on EasyVideoR1 (Qin et al., 2026) with a batch size of 256 and $K = 8$ rollouts per prompt. For ReMind, we set the review ratio $r = 0.10$, review period $f = 5$, and enqueue probability $p_{\text{add}} = 0.25$, so that review samples constitute $\sim$2% of total training prompts. Full training hyperparameters and baseline descriptions are in Appendix D.

4.2 Main Results

Table 1 summarizes the main results. **Replay methods improve sample efficiency but leave retention unaddressed.** The three replay baselines (ExGRPO, RLEP, RePO) improve over GRPO by reusing successful trajectories, with RePO achieving the highest average (62.02). However, they store fixed historical rollouts and cannot detect whether a previously mastered sample has regressed. This limitation surfaces on ZeroBench, where all three replay methods score below GRPO, suggesting that stale trajectories do not prevent forgetting on the hardest tail of the distribution.

ReMind yields larger gains on math-heavy benchmarks and complements online filtering. ReMind outperforms the best replay baseline by +2.04 on average, with notably larger improvement on math reasoning (Math Avg +5.67) than on general benchmarks (General Avg +1.94). This asymmetry is consistent with the correct-set turnover perspective: longer reasoning chains are more fragile under policy drift and thus more prone to regression. The gain is consistent across both GRPO (+3.81) and DAPO (+3.44) backbones, confirming that retention-aware review and online filtering combine effectively.

Table 1: Main results on eight image-text reasoning benchmarks. All methods use Qwen3-VL-8B-Instruct as the base policy. Best results are in **bold** and second-best are underlined. △ rows show the gain over the corresponding base method.

Method	Math Reasoning					General Multimodal					Avg
	Vista	Vision	WeMath	Verse	Avg	MMMU-P	MMB	MM-S	ZeroB	Avg	
Base Model	73.80	47.37	54.10	54.61	57.47	51.75	89.79	71.83	19.76	58.28	57.88
ExGRPO	77.30	55.46	62.67	56.89	63.08	55.49	90.44	72.00	19.01	59.24	61.16
RLEP	78.50	54.23	62.48	58.91	63.53	55.38	90.45	72.27	19.61	59.43	61.48
RePO	78.00	53.77	63.52	59.39	63.67	54.95	90.45	72.60	23.50	60.38	62.02
GRPO	76.20	48.82	56.57	59.52	60.28	55.78	90.29	72.20	22.60	60.22	60.25
GRPO + ReMind	79.10	56.81	66.38	61.52	65.95	**58.78**	90.41	73.40	26.05	62.16	64.06
△	+2.90	+7.99	+9.81	+2.00	+5.67	+3.00	+0.12	+1.20	+3.45	+1.94	+3.81
DAPO	77.60	52.78	57.43	59.55	61.84	56.85	90.53	72.77	20.66	60.20	61.02
DAPO + ReMind	**79.85**	**57.32**	**68.29**	60.37	**66.46**	57.89	**91.22**	**74.70**	**26.05**	**62.47**	**64.46**
△	+2.25	+4.54	+10.86	+0.82	+4.62	+1.04	+0.69	+1.93	+5.39	+2.27	+3.44

4.3 Cross-Modal Generalization

Table 2: Generalization of ReMind to video understanding and text math reasoning. Panel (a) trains on video data with Qwen3-VL-8B-Instruct; panel (b) trains on text math data with Qwen2.5-7B-Math (Yang et al., 2024). Best results are in **bold**.

(a) Video Understanding

Method	MVB	Motion	MLVU	Holmes	LVR	VMQA	Avg
Base Model	69.50	56.10	**78.40**	45.30	77.30	47.80	62.40
GRPO	70.75	57.01	75.80	44.80	78.61	50.33	62.88
ReMind	**71.22**	**61.44**	77.37	**50.84**	**82.49**	**53.19**	**66.09**

(b) Text Math Reasoning

Method	AIME24	AIME25	AMC	MATH	Minerva	Olympiad	Avg
Qwen2.5-7B-Math	12.50	10.20	48.50	80.40	32.70	41.00	37.55
+ SFT	22.20	22.30	52.80	82.60	40.80	43.70	44.07
+ SFT + RL	25.80	23.10	62.70	87.20	39.70	50.40	48.15
+ ExGRPO	32.30	25.70	65.60	87.60	**40.10**	57.00	51.38
+ ReMind	**33.85**	**26.25**	**67.85**	**90.80**	38.97	**57.93**	**52.61**

To test whether the benefits of ReMind extend beyond multimodal image reasoning, we apply it to two additional domains (Table 2). For video understanding, we evaluate on six benchmarks covering general perception (MVBench (Li et al., 2024b), MotionBench (Hong et al., 2025)), long-video comprehension (MLVU (Zhou et al., 2025)), video reasoning (Video-Holmes (Cheng et al., 2025), LongVideo-Reason (Chen et al., 2025)), and STEM knowledge (VideoMathQA (Rasheed et al., 2025)); further details are in Appendix C. For text math reasoning, we evaluate on AIME 2024/2025, AMC (Li et al., 2024a), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024).

ReMind improves video reasoning, especially on reasoning-intensive benchmarks. ReMind raises the video average from 62.88 to 66.09 (+3.21), with gains concentrating on reasoning-heavy tasks (Video-Holmes +6.04, LongVideo-Reason +3.88). Notably, GRPO *drops* 2.6 points below the base model on MLVU, a clear instance of correct-set regression that ReMind partially repairs.

ReMind outperforms replay on text math and generalizes across modalities. On text math (panel b), ReMind surpasses ExGRPO by +1.23 on average, with consistent gains across competition-level benchmarks. Across both settings, the pattern matches the image results: larger gains on tasks with longer reasoning chains, where correct-set turnover is most damaging. This consistency across three modalities confirms that the review mechanism addresses a modality-agnostic failure mode of RLVR training.

4.4 Review Budget Sensitivity

The review budget is controlled by the review ratio r and period f; Table 3 and Figure 4 vary both to study sensitivity.

The method is robust across a wide budget range. All configurations outperform GRPO by a wide margin (62.83–64.06 vs. 60.25). Even 1% review budget is sufficient to break the acquisition–regression stalemate that causes GRPO's mid-training plateau, while performance declines slightly above 4% as over-reviewing displaces fresh training signal. At a matched 2% budget, burst scheduling ($r = 0.10$, $f = 5$) outperforms smooth ($r = 0.02$, $f = 1$) by +0.36, likely because accumulating reviews over several steps produces a more diverse cohort that better distinguishes regressed from stable samples.

Table 3: Sensitivity to review budget and schedule. Shaded row is the default configuration. Full per-benchmark breakdown is in Appendix A.

Method	r	f	Budget	Math	Gen.	Avg
GRPO	–	–	0%	60.28	60.22	60.25
ReMind	0.10	5	∼2%	65.95	62.16	64.06
Budget variants						
weak	0.05	5	∼1%	64.28	61.67	62.98
strong	0.20	5	∼4%	64.50	61.88	63.19
aggressive	0.10	1	∼10%	64.33	61.33	62.83
Schedule variant (matched ∼2% budget)						
smooth	0.02	1	∼2%	65.44	61.95	63.70

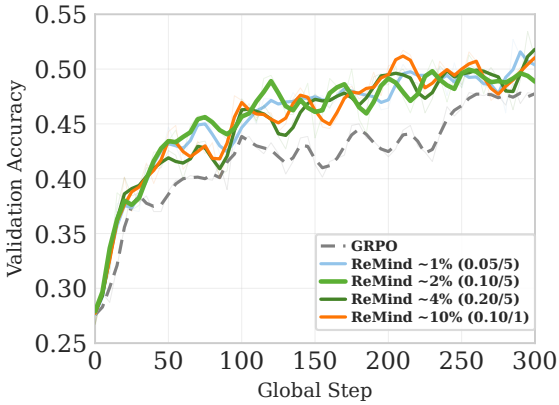


Figure 4: Validation accuracy during training for GRPO and four ReMind budget configurations. All ReMind variants separate from GRPO by step 50 and avoid the mid-training plateau visible in the baseline.

4.5 Training Dynamics

Figure 5 decomposes training into four diagnostic signals. **ReMind improves accuracy by reducing forgetting, not by accelerating acquisition.** Panels (a–b) show that both methods acquire new solutions at nearly the same rate (∼500 ever-solved samples by step 450), yet ReMind maintains a consistent accuracy advantage (0.52 vs. 0.47). The gap is explained by panel (c): GRPO accumulates ∼190 forgotten samples, whereas ReMind stabilizes around 155, a reduction of ∼35 samples. Notably, ∼155 samples still regress even with review, suggesting that some degree of correct-set turnover is intrinsic to the RLVR optimization process; further reducing this residual forgetting is a promising direction for future work.

Review mitigates entropy collapse. Panel (d) reveals a temporal correspondence between entropy decline and forgetting onset in GRPO: both accelerate between steps 50 and 150, suggesting that as the policy concentrates probability mass on fewer

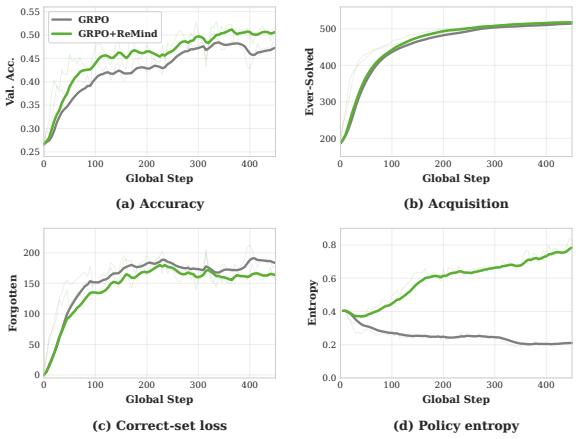


Figure 5: Training dynamics of GRPO and GRPO+ReMind over 450 steps. (a) Validation accuracy on a held-out subset. (b) Cumulative number of distinct samples ever solved (acquisition). (c) Number of previously solved samples that the current policy fails to solve (correct-set loss). (d) Policy entropy averaged over generated tokens.

generation paths, it loses the ability to reproduce previously correct reasoning chains. ReMind reverses the initial entropy drop and sustains entropy near 0.7–0.8, indicating that periodic re-exposure to mastered samples counteracts this concentration and preserves diverse solution strategies.

4.6 Further Analysis

Targeted review outperforms naive continued training. A natural alternative to the review mechanism is simply training longer. Figure 6 compares ReMind (one epoch) against GRPO trained for two full epochs on the same data. The second epoch yields minimal improvement: accuracy oscillates around 0.50 and finishes at 0.497 (+2.0 pp) despite doubling the training budget. In contrast, ReMind matches the two-epoch peak accuracy at step 296 vs. step 740 (2.5× faster), and ultimately achieves 0.539 (+4.1 pp above the two-epoch endpoint). The instability of the second-epoch curve, alternating between brief gains and regressions, is consistent with naive re-exposure causing repeated overfitting rather than stable retention, confirming that selective, policy-aware review cannot be replaced by simply training longer.

Correct-set turnover is persistent, and the review queue progressively reduces it. Figure 7 tracks queue behavior across 79 review events. The per-event retention rate rises from 72% early (steps 50–150) to 82% late (steps 300–440), while the number of regressed samples per event drops from

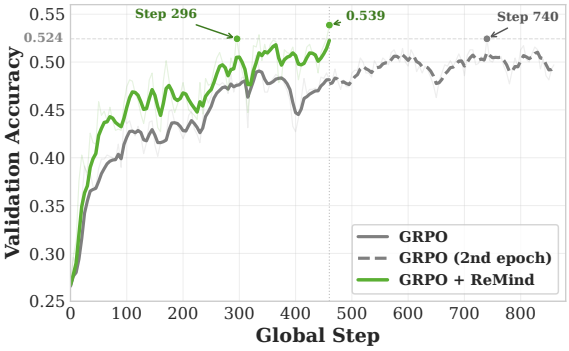
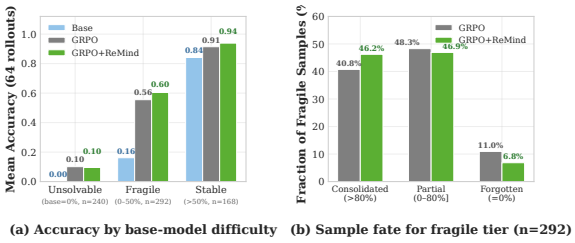


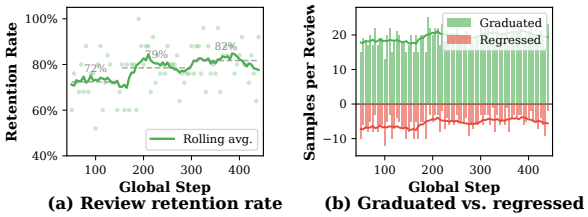
Figure 6: Validation accuracy for GRPO trained for one epoch, GRPO trained for two full epochs on the same data, and GRPO+ReMind (one epoch).



(a) Accuracy by base-model difficulty (b) Sample fate for fragile tier (n=292)

Figure 8: Per-difficulty analysis on 700 validation samples, each rolled out 64 times under the base model, GRPO, and GRPO+ReMind. Samples are stratified by base-model mean accuracy into three tiers. (a) Mean accuracy per tier. (b) Outcome distribution for fragile-tier samples after training. A detailed per-bin breakdown is in Appendix B.

6.9 to 4.6, indicating that the self-correcting review cycle gradually stabilizes mastery. However, regression never vanishes entirely: even in the final phase, 4–5 samples per event still drift below full mastery, confirming that correct-set turnover is intrinsic to the RLVR optimization rather than a transient early-training artifact.



(a) Review retention rate (b) Graduated vs. regressed

Figure 7: Review queue dynamics over training. (a) Retention rate at each review event (dots) with rolling average (line) and per-phase means (dashed). (b) Number of graduated (confirmed retained, removed from queue) and regressed (re-enqueued for future repair) samples per review event.

Retention gains concentrate on fragile samples that are solvable but unstable. Figure 8 stratifies 700 validation samples into three tiers by base-model accuracy: *unsolvable* (0%, n=240), *fragile* (0–50%, n=292), and *stable* (>50%, n=168). ReMind's improvement over GRPO is largest on the fragile tier (+5.1 pp) and negligible on unsolvable samples, consistent with the premise that review can only retain previously acquired knowledge. Within the fragile tier, ReMind consolidates 46.2% of samples to above 80% accuracy (vs. 40.8% under GRPO) and reduces complete forgetting from 11.0% to 6.8%, shifting fragile knowledge toward stable mastery rather than catastrophic loss.

The review mechanism adds negligible computational overhead. As shown in Table 4, the review mechanism accounts for only 3% of total wall-clock time. No additional model replicas, optimizer states, or experience buffers are required beyond the lightweight FIFO queue, making ReMind straightforward to integrate into any RLVR training loop.

Table 4: Wall-clock breakdown of a GRPO+ReMind training run. Review overhead is measured as the marginal step-time increase on review steps relative to non-review steps.

Component	Time	Fraction
Standard training (rollout + update)	9.9 h	78%
Periodic validation	2.5 h	19%
Review (re-rollout + queue mgmt.)	0.3 h	**3%**
Total	12.7 h	100%

5 Conclusion

We identify *correct-set turnover* as a systematic cause of accuracy plateaus in RLVR, and show that timely review within a low-cost repair window can effectively counter it. ReMind operationalizes this principle with a lightweight, self-correcting review queue; with only 3% additional cost, it yields consistent accuracy gains across image, video, and text reasoning with different modalities, models, and algorithmic backbones. More broadly, our results suggest that a meaningful share of the headroom in RLVR is not only about learning more, but also about losing less. Residual turnover persists even with review, indicating that some regression is intrinsic to RLVR optimization; we hope this work encourages the community to treat sample-level retention as a primary design axis in reasoning RL, alongside reward design, exploration, and curriculum that make forgetting visible and addressable.

Limitations

ReMind inherits the standard RLVR assumption that the verifiable reward is correct; because mastery status directly governs which samples enter the review queue, systematic reward errors (e.g., false positives from imprecise answer extraction) would be more consequential here than in methods that do not condition on mastery. Our experiments cover 7–8B-scale models on mathematical and multimodal reasoning; whether the optimal review budget or the magnitude of retention gains transfers to substantially larger models or other domains such as code generation remains to be verified. ReMind reduces but does not eliminate correct-set regression; our current design uses a fixed review schedule, and adaptive scheduling that responds to the model's regression dynamics could further narrow this gap.

Ethics Statement

This work proposes a training-time review mechanism for reinforcement learning with verifiable rewards; it does not introduce new data collection, human annotation, or model deployment, and all experiments use publicly available models and benchmarks. ReMind is designed to improve training stability and does not alter the model's safety properties beyond what the underlying RLVR training already provides. Because the method prioritizes retaining previously mastered samples, practitioners should monitor generalization when applying it to narrow training distributions to avoid overfitting to dataset-specific patterns.

Use of AI Assistants

We used AI Assistants to assist with writing polishing, including grammar checking and sentence-level rewording. All technical content, experimental design, and scientific claims were produced by the authors. The AI assistant was not used for data analysis, code generation, or result interpretation.

References

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Nicholas J Cepeda, Harold Pashler, Edward Vul, John T Wixted, and Doug Rohrer. 2006. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132(3):354.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. 2025. Scaling rl to long videos. *Preprint*, arXiv:2507.07966.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. 2025. Video-holmes: Can mllm think like holmes for complex video reasoning? *Preprint*, arXiv:2505.21374.

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

Yiming Dong, Kun Fu, Haoyu Li, Xinyuan Zhu, Yurou Liu, Lijing Shao, Jieping Ye, and Zheng Wang. 2026. Probing RLVR training instability through the lens of objective-level hacking. *arXiv preprint arXiv:2602.01103*.

Shihan Dou, Muling Wu, Jingwen Xu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025. Improving RL exploration for LLM reasoning through retrospective replay. *arXiv preprint arXiv:2504.14363*.

Hermann Ebbinghaus. 1885. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. 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)*, pages 3828–3850.

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

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. 2025. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. *Preprint*, arXiv:2501.02955.

Sihan Hu, Xiansheng Cai, Yuan Huang, Zhiyuan Yao, Linfeng Zhang, Pan Zhang, Youjin Deng, and Kun Chen. 2026. Emergent slow thinking in llms as inverse tree freezing. *Preprint*, arXiv:2509.23629.

Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. 2025. VCRL: Variance-based curriculum reinforcement learning for large language models. *arXiv preprint arXiv:2509.19803*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024a. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. *Preprint*, arXiv:2311.17005.

Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. 2025. Repo: Replay-enhanced policy optimization. *arXiv preprint arXiv:2506.09340*.

Honglin Lin, Zheng Liu, Yun Zhu, Chonghan Qin, Juekai Lin, Xiaoran Shang, Conghui He, Wentao Zhang, and Lijun Wu. 2026. Mmfinereason: Closing the multimodal reasoning gap via open data-centric methods. *Preprint*, arXiv:2601.21821.

Long-Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

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 *The Twelfth International Conference on Learning Representations*.

Weiyu Ma, Yongcheng Zeng, Yan Song, Xinyu Cui, Jian Zhao, Xuhui Liu, and Mohamed Elhoseiny. 2026. Freshness-aware prioritized experience replay for LLM/VLM reinforcement learning. *arXiv preprint arXiv:2604.16918*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, and 1 others. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Hoang Phan, Xianjun Yang, Kevin Yao, Jingyu Zhang, Shengjie Bi, Xiaocheng Tang, Madian Khabsa, Lijuan Liu, and Deren Lei. 2025. Beyond reasoning gains: Mitigating general capabilities forgetting in large reasoning models. *arXiv preprint arXiv:2510.21978*.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, and 1 others. 2025. 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)*, pages 20023–20070.

Chuanyu Qin, Chenxu Yang, Qingyi Si, Naibin Gu, Dingyu Yao, Zheng Lin, Peng Fu, Nan Duan, and Jiaqi Wang. 2026. Easyvideor1: Easier rl for video understanding. *Preprint*, arXiv:2604.16893.

Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad Shahbaz Khan. 2025. Videomathqa: Benchmarking mathematical reasoning via multimodal understanding in videos. *Preprint*, arXiv:2506.05349.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. *Preprint*, arXiv:1611.07725.

Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, and 1 others. 2025. Zerobench: An impossible visual benchmark for contemporary large multimodal models. *arXiv preprint arXiv:2502.09696*.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, Daya Guo, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Mariya Toneva, Alessandro Sordoni, Rémi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.

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

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. DAPO: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025b. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. 2025. ExGRPO: Learning to reason from experience. *Preprint*, arXiv:2510.02245.

Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. 2025a. RLEP: Reinforcement learning with experience replay for llm reasoning. *arXiv preprint arXiv:2507.07451*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Yuheng Zhang, Wenlin Yao, Changlong Yu, Yao Liu, Qingyu Yin, Bing Yin, Hyokun Yun, and Lihong Li. 2025b. Improving sampling efficiency in RLVR through adaptive rollout and response reuse. *arXiv preprint arXiv:2509.25808*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025. Mlvu: Benchmarking multi-task long video understanding. *Preprint*, arXiv:2406.04264.

A Full Budget Sensitivity Results

Table 5 reports per-benchmark results for all configurations discussed in §4.4.

Budget analysis. Across the four budget levels, math reasoning performance varies more than general multimodal performance (Math Avg spans 64.28–65.95 vs. General Avg 61.33–62.16), confirming that the review mechanism primarily benefits reasoning-intensive tasks where correct-set turnover is more frequent. The 10% budget ($f = 1$) underperforms the 4% budget despite allocating more capacity to review: at every step, 10% of the batch consists of previously mastered samples, reducing the diversity of fresh training signal. This suggests a trade-off between retention and acquisition that is well balanced at the 2% default.

Schedule analysis. The smooth schedule (reviewing 2% every step) achieves a slightly higher General Avg (61.95 vs. 62.16) but lower Math Avg (65.44 vs. 65.95) compared to burst. One explanation is that frequent small reviews maintain general capabilities through steady reinforcement, while burst reviews are more effective at catching and repairing regressions on harder reasoning samples that drift faster between review steps.

Table 5: Full per-benchmark results for the budget and schedule sensitivity study (§4.4). The upper block varies the review budget at fixed periodic schedule; the lower block compares burst and smooth schedules at matched ∼2% budget. Shaded rows indicate the default configuration.

Setting	r	f	Math Reasoning					General Multimodal					Avg
			Vista	Vision	WeMath	Verse	Avg	MMMU-P	MMB	MM-S	ZeroB	Avg	
Budget ablation													
∼1%	0.05	5	78.45	55.20	63.33	60.15	64.28	57.76	90.68	73.10	25.15	61.67	62.98
∼2% (default)	0.10	5	79.10	56.81	66.38	61.52	65.95	58.78	90.41	73.40	26.05	62.16	64.06
∼4%	0.20	5	76.50	55.46	66.00	60.05	64.50	57.58	90.84	73.20	25.90	61.88	63.19
∼10%	0.10	1	76.25	55.46	65.24	60.37	64.33	57.43	90.78	72.27	24.85	61.33	62.83
Schedule ablation (matched ∼2% budget)													
Burst	0.10	5	79.10	56.81	66.38	61.52	65.95	58.78	90.41	73.40	26.05	62.16	64.06
Smooth	0.02	1	78.45	57.27	64.86	61.17	65.44	57.58	90.75	74.03	25.45	61.95	63.70

B Per-Difficulty Forgetting Breakdown

Table 6 provides a fine-grained breakdown of the per-difficulty analysis summarized in §4.6. We roll out each of the 700 validation samples 64 times under the base model (Qwen3-VL-8B-Instruct), vanilla GRPO, and GRPO+ReMind, and stratify samples into seven bins by the base model's mean accuracy.

Observations. The improvement from ReMind over GRPO is non-uniform across difficulty levels. Samples in the $(30, 70\%]$ range benefit most (accuracy gain of +6.4–6.7 pp over GRPO), while unsolvable samples ($= 0\%$) show no meaningful difference. This pattern is expected: unsolvable samples never enter the correct set, so the review mechanism has no mastery to protect; easy samples ($> 90\%$) are already robust to policy drift and rarely regress. The fragile middle ground, where the base model sometimes produces a correct solution but does so inconsistently, is precisely where correct-set turnover is concentrated and where targeted review provides the most value.

Across all bins with nonzero base accuracy, ReMind consistently reduces the fraction of regressed samples: from 24.1% to 16.8% in the $(0, 10\%]$ bin and from 21.8% to 14.5% in the $(30, 50\%]$ bin. The effect is smallest for the $(90, 100\%]$ bin, where both methods maintain near-perfect accuracy and regression is rare regardless.

Mastery consolidation. Among the 292 fragile-tier samples ($0 <$ base accuracy $\leq 50\%$), GRPO consolidates 119 (40.8%) to above 80% accuracy, while ReMind consolidates 135 (46.2%). Conversely, GRPO completely forgets 32 samples (11.0%), compared with 20 (6.8%) under ReMind. This consolidation effect explains much of ReMind's advantage on the downstream benchmarks: by preventing fragile knowledge from decaying to zero and instead reinforcing it toward stable mastery, the review mechanism converts partial solutions into reliable performance.

C Video Understanding Benchmarks

Table 7 summarizes the six video understanding benchmarks used in §4.3.

D Experimental Setup Details

Training data. MMFineReason-123K (Lin et al., 2026) is derived from the MMFineReason-1.8M corpus via difficulty-based filtering. Each candidate sample is rolled out four times with Qwen3-VL-4B-Thinking (Bai et al., 2025), and only samples on which the model fails every attempt are retained, yielding a training set of 123K challenging multimodal reasoning problems.

Evaluation benchmarks. We evaluate on eight multimodal reasoning benchmarks spanning visually grounded mathematics, broad subject knowledge, and fine-grained visual understanding. MathVista (Lu et al.), MathVision (Wang et al., 2024), WeMath (Qiao et al., 2025), and MathVerse (Zhang et al., 2024) target mathematical reasoning in visual contexts, from general problem solving to competition-level diagrams and structured difficulty tiers. MMMU-Pro (Yue et al., 2025a) and MMBench (Liu et al., 2024) test broad multi-discipline knowledge and general multimodal competence. MM-Star (Chen et al., 2024) stresses fine-grained visual discrimination, and ZeroBench (Roberts et al., 2025) serves as a hard stress test for the tail of the reasoning distribution.

Table 6: Fine-grained per-difficulty analysis. Each row corresponds to a bin of validation samples grouped by the base model's mean accuracy over 64 rollouts. **Mean Acc.** is the average accuracy within each bin; **Regressed** counts samples whose accuracy dropped by more than 5 pp relative to the base model.

Base Accuracy Bin	n	**Mean Acc.**			**Δ vs. GRPO**		**Regressed (%)**	
		Base	GRPO	ReMind	Δ(Acc.)	Δ(Regr.)	GRPO	ReMind
$= 0\%$	240	0.000	0.101	0.096	−0.006	0.0	0.0	0.0
$(0, 10\%]$	137	0.047	0.443	0.479	+0.036	−7.3	24.1	16.8
$(10, 30\%]$	100	0.187	0.630	0.689	+0.059	−4.0	20.0	16.0
$(30, 50\%]$	55	0.403	0.699	0.763	+0.064	−7.3	21.8	14.5
$(50, 70\%]$	43	0.591	0.835	0.902	+0.067	−7.0	16.3	9.3
$(70, 90\%]$	38	0.806	0.859	0.883	+0.024	−2.6	21.1	18.4
$(90, 100\%]$	87	0.981	0.978	0.981	+0.004	+1.1	12.6	13.8
All	700	0.269	0.486	0.510	+0.024	−2.3	7.7	5.4

Table 7: Video understanding benchmarks used in the cross-modal evaluation (§4.3).

Benchmark	**Samples**	**Metric**
General Video Understanding		
MVBench (Li et al., 2024b)	3,586	Accuracy
MotionBench (Hong et al., 2025)	3,715	Accuracy
Long Video Understanding		
MLVU (Zhou et al., 2025)	502	Accuracy
Video Reasoning		
Video-Holmes (Cheng et al., 2025)	1,837	Accuracy
LongVideo-Reason (Chen et al., 2025)	851	Accuracy
STEM Knowledge		
VideoMathQA (Rasheed et al., 2025)	2,100	Accuracy

Baselines. All methods are implemented on a shared GRPO-style backbone so that performance differences mainly reflect the retention or replay strategy rather than optimization differences. GRPO (Shao et al., 2024) is the pure on-policy reference. DAPO (Yu et al., 2025) augments GRPO with dynamic sampling and online filtering. Ex-GRPO (Zhan et al., 2025) reuses successful trajectories from historical replay buffers. RLEP (Zhang et al., 2025a) replays trajectories from a far-future checkpoint. RePO (Li et al., 2025) collects early on-policy rollouts and revisits them asynchronously. We also report the base model (Qwen3-VL-8B-Instruct) without RL post-training as a reference point.

Implementation details. Our implementation is built on the EasyVideoR1 (Qin et al., 2026) framework. The maximum context length is 8192 tokens, split into a prompt budget of 4096 and a response budget of 4096, applied consistently at training and evaluation time. We train with a learning rate of 1×10^{-6} and a batch size of 256, sampling $K = 8$ rollouts per prompt at temperature 1.0. Clipping thresholds are set to $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$; we omit both the KL penalty and entropy regularization from the objective. For ReMind, we set the review ratio $r = 0.10$, review period $f = 5$, and enqueue probability $p_{\text{add}} = 0.25$. Under these settings, review samples constitute approximately 2% of total training prompts on average. All experiments are run on 4 compute nodes, each with 8 NVIDIA H200 140GB GPUs.