TRAIN WHAT IS READY BUT NOT YET MASTERED: COMPETENCE-DRIVEN CURRICULUM SAMPLING FOR REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) is an effective paradigm for improving mathematical reasoning in large language models, yet the training prompt distribution is commonly determined by random sampling or manually specified curricula. Such schedules ignore that the useful training difficulty changes with the learner's competence. We introduce a competence-driven curriculum sampler that dynamically allocates each training batch across ordered difficulty regions according to two complementary signals: *cumulative prerequisite readiness*, which estimates whether the model has acquired the easier abilities required to approach a region, and *mastery gap*, which measures how much competence remains to be learned in that region. We maintain an exponentially smoothed success rate for each difficulty bucket, define readiness as the product of competence estimates over all prerequisite buckets, and assign sampling priority proportional to readiness times remaining mastery gap. This yields a closed-loop curriculum whose progression is driven directly by model competence rather than training step. On DeepMath-10K, the same scheduler achieves the highest eight-benchmark mean among the compared sampling baselines at both Qwen3-1.7B and Qwen3-4B scales: 0.3992 versus 0.3944 for the strongest 1.7B baseline, and 0.5617 versus 0.5485 for the strongest 4B baseline. Difficulty-range ablations further show that the best fixed range changes with model scale, while our adaptive scheduler remains best at both scales. These results support a simple principle for RLVR data selection: *train what is ready to learn, but not yet mastered*.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become an important approach for improving mathematical reasoning in large language models (LLMs). Given automatically verifiable answers, policy optimization algorithms such as GRPO (Shao et al., 2024) can optimize reasoning trajectories directly from outcome rewards. However, while substantial effort has focused on reward design, advantage estimation, rollout efficiency, and policy optimization, comparatively less attention has been paid to a basic question: *which training problems should the model see at each stage of reinforcement learning?*

A standard strategy is random sampling, while curriculum learning emphasizes easier examples earlier and harder examples later. However, fixed easy-to-hard schedules may mismatch the learner: useful regions can already be mastered at one checkpoint but remain inaccessible at another.

We argue that curriculum progression should therefore be determined by *model competence rather than training time*. A useful difficulty region should satisfy two conditions. First, the prerequisite abilities needed to approach the region should already be sufficiently developed. Second, the target region itself should still contain substantial room for learning. This motivates the principle

Train what is ready to learn, but not yet mastered.

Based on this principle, we introduce a lightweight bucket-level adaptive curriculum for RLVR. We partition the training set into ordered difficulty buckets and maintain an online estimate of model competence for each bucket. We decompose the sampling priority of bucket k into

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.

$$\text{Priority}_k = \underbrace{\text{Prerequisite Readiness}_k}_{\text{can learn}} \times \underbrace{\text{Mastery Gap}_k}_{\text{still needs learning}}. \quad (1)$$

Our main design is *cumulative prerequisite readiness*. We define the readiness of each harder bucket directly from the current competence of all easier prerequisite buckets, allowing curriculum progression to emerge continuously from the learner's state.

Our method modifies only data sampling. The reward function, rollout procedure, GRPO objective, advantage estimator, and optimizer remain unchanged. This makes the approach complementary to improvements in RL algorithms themselves.

Our contributions are:

- We formulate adaptive RLVR sampling as a *readiness–mastery* curriculum allocation problem, introducing cumulative prerequisite readiness to estimate difficulty accessibility from current model competence and mastery gaps.
- We characterize the scheduler's structural properties, showing that improving prerequisite competence monotonically increases the priority of harder regions, while increasing target-region mastery reduces their priority.
- Extensive experiments on Qwen3-1.7B and Qwen3-4B demonstrate improved aggregate performance over existing sampling baselines across eight mathematical reasoning benchmarks, while difficulty-range ablations reveal that adaptive allocation generalizes across model scales beyond fixed curriculum ranges.

2 Related Work

2.1 Reinforcement Learning for Mathematical Reasoning

Explicit chain-of-thought prompting substantially improves multi-step reasoning in large language models (Wei et al., 2022), and zero-shot variants show that step-by-step reasoning can also be elicited without task-specific demonstrations (Kojima et al., 2022). In RL-based mathematical reasoning, DeepSeekMath introduced Group Relative Policy Optimization (GRPO) (Shao et al., 2024), while systems such as DeepSeek-R1 (DeepSeek-AI, 2025) and DAPO (Yu et al., 2025) demonstrate the effectiveness of large-scale reinforcement learning with verifiable outcome rewards. Process-level verification provides a complementary route to improving mathematical reasoning (Lightman et al., 2024).

Several recent works study the data-selection problem in reasoning RL. Online Difficulty Filtering (ODF) selects problems with intermediate online success rates and provides a theoretical argument for why such filtering can improve RLVR efficiency (Bae et al., 2026). CDAS maintains historical, per-problem difficulty estimates and samples problems whose estimated difficulty aligns with a global model-competence variable (Kong et al., 2025). CROPI uses off-policy influence estimates to identify training data that are useful to the current policy (Zhu et al., 2026), while LPPO uses learning-progress signals for sample-centric optimization (Chen et al., 2026). Large-scale empirical studies further show that model scale, data volume, and RL compute interact strongly during mathematical post-training (Tan et al., 2026). Our method is complementary to these approaches: it leaves the RL objective unchanged and adapts the prompt distribution through an ordered, bucket-level competence state.

2.2 Curriculum and Self-Paced Learning

Curriculum learning studies how ordering examples from easier to harder can improve optimization and generalization (Bengio et al., 2009). Self-paced learning lets the learner state influence example selection (Kumar et al., 2010), and data-driven curricula have also been learned from training dynamics (Jiang et al., 2018). In NLP, competence-based curriculum learning adapts the accessible training distribution to estimated learner competence (Platanios et al., 2019). More recent work studies curriculum design for language-model training and mathematical reasoning at substantially larger scales (Zhang et al., 2026; Jia et al., 2026).

For reasoning post-training, Automatic Curriculum Expert Iteration (Auto-CEI) adapts a curriculum to the model's evolving reasoning capability (Zhao et al., 2025). E2H Reasoner studies curriculum RLVR directly and introduces classical, balanced, cosine, and Gaussian easy-to-hard schedulers (Parashar et al., 2026). The latter is particularly relevant to our static curriculum baselines. E2H schedules difficulty as a function of training progress, whereas our sampler makes the allocation a

function of the model's measured competence. This distinction is motivated by evidence that the preferred curriculum can depend on model capability and task difficulty (Jia et al., 2026).

2.3 Adaptive Difficulty Sampling

The closest adaptive methods differ primarily in the state they maintain and the criterion they optimize. ODF favors problems with intermediate current success rates (Bae et al., 2026); CDAS aligns per-problem historical difficulty with a global competence scalar (Kong et al., 2025); and learning-progress or influence-based methods prioritize samples according to estimated optimization value (Chen et al., 2026; Zhu et al., 2026). In contrast, our scheduler operates over ordered difficulty regions and explicitly decomposes sampling priority into *prerequisite readiness* and *remaining mastery need*. With K buckets, it maintains only $O(K)$ online competence states and asks a different question: *which difficulty region is ready to receive more training budget, while still remaining insufficiently mastered?*

3 Method

3.1 Problem Setup

We partition the RLVR dataset into K difficulty buckets:

$$\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{B}_k,$$

ordered from easy to hard. For DeepMath-10K, we use

$$\mathcal{B}_1 = \mathrm{D3_4}, \quad \mathcal{B}_2 = \mathrm{D5}, \quad \mathcal{B}_3 = \mathrm{D6}, \quad \mathcal{B}_4 = \mathrm{D7} \cup \mathrm{D8_9}.$$

At step t, the sampler selects $\mathbf{q}_t = [q_{1,t}, \ldots, q_{K,t}]$ satisfying $\sum_k q_{k,t} = B$, where allocations are determined by online competence rather than predefined training schedules.

3.2 Calibration and Bucket-Level Competence

Before RL training, we perform a no-gradient calibration rollout for each bucket. Let $\mathcal{C}_k \subset \mathcal{B}_k$ denote the calibration subset for bucket k. Given rollout group size G, its empirical success rate is

$$p_k^{\mathrm{cal}} = \frac{\#\text{ correct rollout responses on } \mathcal{C}_k}{\#\text{ rollout responses evaluated on } \mathcal{C}_k}. \tag{2}$$

We initialize the online competence state as

$$\bar{p}_{k,0} = p_k^{\mathrm{cal}}, \qquad \bar{\mathbf{p}}_0 = [\bar{p}_{1,0}, \ldots, \bar{p}_{K,0}], \tag{3}$$

rather than assigning all buckets the same initial value.
During RL training, bucket competence is updated using an exponential moving average:

$$\bar{p}_{k,t} = (1-\alpha)\bar{p}_{k,t-1} + \alpha p_{k,t}, \tag{4}$$

where $p_{k,t}$ is the success rate observed from current on-policy rollouts for bucket k. EMA reduces sensitivity to stochastic rollout fluctuations while retaining online adaptation to policy improvement.

3.3 Cumulative Prerequisite Readiness

A large mastery gap alone should not imply that a difficult region deserves more training. If the model has not yet developed the prerequisite abilities, repeatedly sampling very difficult problems may be inefficient.
We define the readiness of the easiest bucket as

$$R_{1,t} = 1. \tag{5}$$

For harder buckets,

$$R_{k,t} = \prod_{j<k} \bar{p}_{j,t-1}, \qquad k > 1. \tag{6}$$

For four buckets,

$$R_1 = 1, \quad R_2 = \bar{p}_1, \quad R_3 = \bar{p}_1\bar{p}_2, \quad R_4 = \bar{p}_1\bar{p}_2\bar{p}_3.$$

The multiplicative form means that harder regions receive high readiness only when prerequisite regions have accumulated competence. Under a simplifying independence assumption, if $\bar{p}_j$ approximates prerequisite availability,

$$R_k \approx P(\text{all prerequisites of bucket } k \text{ are mastered}). \tag{7}$$

Equation 6 uses the competence state available before step t to determine the batch for step t. As competence evolves, the readiness of harder regions changes continuously with the learner state.

3.4 Mastery Gap

Readiness determines whether a region is currently approachable, but not whether it still requires training. We therefore define the mastery gap as

$$M_{k,t} = 1 - \bar{p}_{k,t-1}. \tag{8}$$

$M_{k,t}$ is large when bucket k remains insufficiently learned and decreases as competence improves. Unlike uncertainty measures such as $4p(1-p)$, it captures remaining learning need rather than outcome ambiguity.

3.5 Readiness–Mastery Priority

We combine readiness and mastery gap multiplicatively:

$$s_{k,t} = R_{k,t}\left(M_{k,t} + \epsilon\right), \tag{9}$$

or equivalently,

$$s_{k,t} = \left(\prod_{j<k} \bar{p}_{j,t-1}\right)\left(1 - \bar{p}_{k,t-1} + \epsilon\right). \tag{10}$$

Thus,

$$\boxed{\text{Priority} = \text{Prerequisite Readiness} \times \text{Remaining Learning Need.}} \tag{11}$$

High priority requires both sufficient prerequisite competence and a substantial remaining mastery gap.

3.6 Dynamic Quota Allocation

Given scores $\{s_{k,t}\}_{k=1}^{K}$, every non-empty bucket first receives a minimum quota $q_{\min}$. The remaining budget $B - Kq_{\min}$ is then allocated approximately in proportion to the normalized scores:

$$\tilde{q}_{k,t} \propto \frac{s_{k,t}}{\sum_j s_{j,t}}. \tag{12}$$

We use a largest-remainder integer allocation so that

$$\sum_k q_{k,t} = B \tag{13}$$

holds exactly. The minimum quota prevents complete starvation of low-priority regions and maintains persistent exploration.

3.7 Online Closed-Loop Curriculum

The complete feedback loop is

$$\text{Rollout} \rightarrow \text{Competence} \rightarrow \text{Readiness} \rightarrow \text{Mastery Gap} \rightarrow \text{Score} \rightarrow \text{Quota}. \tag{14}$$

The resulting quota creates a closed loop between model learning and data selection. The method changes only prompt sampling; reward computation, rollout generation, GRPO objective, and optimizer remain unchanged.

A complete pseudocode description of the training procedure is provided in Appendix A. The appendix version makes explicit the calibration stage, quota computation, GRPO update, and online competence feedback used by the scheduler. The realized quota trajectories are presented in Appendix A as a compact two-panel visualization.

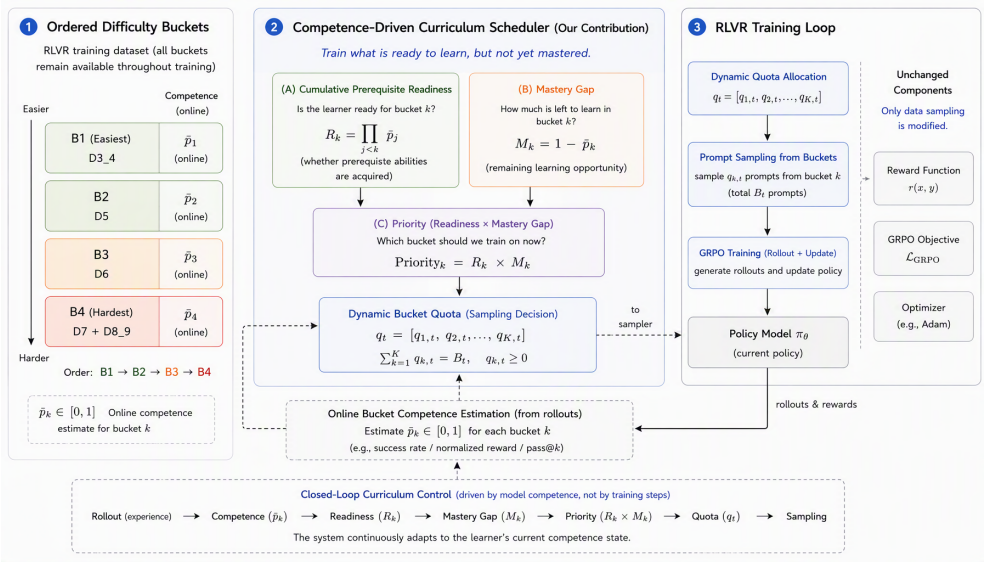


Figure 1: **Overview of the competence-driven curriculum sampler.** Online bucket competence determines cumulative prerequisite readiness and mastery gap; their product defines the sampling priority, which is converted into the per-bucket quota for the next RLVR batch.

4 Theoretical Properties

The score in Eq. 9 satisfies two monotonicity properties that formalize the intended curriculum behavior.

Proposition 1 (Prerequisite monotonicity). For any prerequisite bucket $j < k$,

$$\frac{\partial s_k}{\partial \bar{p}_j} = \left(\prod_{\substack{\ell<k \\ \ell \neq j}} \bar{p}_\ell \right) (1 - \bar{p}_k + \epsilon) \geq 0. \tag{15}$$

Thus improving prerequisite competence can only increase, or leave unchanged, the priority of a harder bucket.

Proposition 2 (Mastery suppression). For the target bucket,

$$\frac{\partial s_k}{\partial \bar{p}_k} = -R_k \leq 0. \tag{16}$$

Thus, as a bucket becomes mastered, its own sampling priority monotonically decreases.
Together, these properties formalize the principle that the scheduler prioritizes regions that are simultaneously ready and insufficiently mastered. Complete proofs and additional properties are given in Appendix E.

5 Experimental Setup

5.1 Training Data

We conduct RLVR training on DeepMath-10K, a 10,000-prompt training subset constructed from DeepMath-103K, following the DeepMath family of verifiable mathematical reasoning data (He et al., 2025). The train/validation/test subsets are created from DeepMath-103K using deterministic cleaning, question-level deduplication, and joint topic–difficulty stratification; full construction details are provided in Appendix B. The original difficulty labels are grouped into four curriculum buckets. Their sizes are

5.2 Models and RL Configuration

We evaluate the same curriculum principle at two model scales, Qwen3-1.7B and Qwen3-4B (Yang et al., 2025). All compared training runs use 156 optimization steps. Training is implemented with

Table 1: Difficulty bucket construction on DeepMath-10K.

Bucket	Raw difficulty	Number of prompts
B1	D3_4	1,393
B2	D5	3,016
B3	D6	2,443
B4	D7 + D8_9	3,148

the veRL/HybridFlow framework (Sheng et al., 2024) and uses GRPO (Shao et al., 2024) with a global prompt batch size of 64 and $G = 8$ rollout responses per prompt. Rollouts are generated with vLLM (Kwon et al., 2023) using a maximum response length of 4096 tokens, temperature 1.0, top-p 1.0, and no top-k truncation. We use AdamW (Loshchilov & Hutter, 2019) with a constant learning rate of 10^{-6}. The clipping ratio is 0.2, the policy mini-batch size is 32, and the actor micro-batch size is 4. We do not add an explicit KL penalty to either the reward or actor loss.

Each difficulty bucket is calibrated with 64 prompts before RL training. This lightweight calibration budget provides an initial competence estimate while introducing negligible overhead compared with the full RL training process. The scheduler is recomputed after every training step, and each non-empty bucket receives at least one prompt per batch. For both Qwen3-1.7B and Qwen3-4B, we use the same scheduler hyperparameters: an EMA coefficient of $\alpha = 0.10$ and a score offset of $\epsilon = 0.10$. No model-specific tuning of these two hyperparameters is performed. The same cumulative-readiness and mastery-gap scheduler, including the same hyperparameter settings, is used at both model scales. Exact launch configurations for all reported runs are included in the supplementary material.

5.3 BASELINES

We compare against random sampling and the easy-to-hard schedule family studied by E2H Reasoner (Parashar et al., 2026). In particular:

- **Random**: random sampling from the full training pool.
- **Balanced**: difficulty-uniform sampling, enforcing equal per-bucket counts in each batch, i.e., $q_1 = \cdots = q_K = B/K$; this corresponds to the balanced E2H control.
- **Classic**: the classical easy-to-hard E2H curriculum, in which the training focus advances from easier to harder difficulty regions according to training progress.
- **Cosine (E2H-C)**: the cosine scheduler of E2H Reasoner, which smoothly shifts sampling mass from easier to harder regions over training.
- **Gaussian (E2H-G)**: the Gaussian scheduler of E2H Reasoner, which uses a moving Gaussian center over ordered difficulty levels to shift the sampling distribution from easy to hard (Parashar et al., 2026).
- **ODF**: Online Difficulty Filtering selects training problems according to their current online success rates, favoring samples with intermediate difficulty that are neither too easy nor too difficult (Bae et al., 2026).
- **CDAS**: CDAS maintains historical difficulty estimates at the problem level and aligns sampled problems with the current model competence state (Kong et al., 2025).

These baselines are step-driven, whereas our method recomputes the distribution from online competence.

We additionally construct a difficulty-range ablation suite to study how the accessible difficulty range affects training. All ablation variants use the full training sample budget, while restricting the available difficulty regions as follows:

- **A1**: training with B4 only.
- **A2**: training with B3–B4.
- **A3**: training with B2–B4.
- **A4**: training with the full B1–B4 difficulty range.

A4 should not be confused with the **Balanced** baseline. A4 samples randomly over the full B1–B4 range within the A1–A4 difficulty-range ablation protocol, whereas Balanced explicitly enforces equal per-bucket counts in every training batch. A4 and the standalone Random baseline have the same nominal full-range random-sampling rule but come from separate experimental runs; we therefore do not interpret their numerical difference as an algorithmic effect.

Table 2: Main results on Qwen3-1.7B and Qwen3-4B across eight mathematical reasoning benchmarks. DeepMath represents the in-domain evaluation benchmark, while the remaining benchmarks measure out-of-domain generalization. All benchmark scores are reported as avg@16, obtained by averaging correctness over 16 independently sampled responses per problem. Classic, Balanced, Cosine, and Gaussian follow the E2H Reasoner schedule family (Parashar et al., 2026). ODF denotes Online Difficulty Filtering, an adaptive sampling baseline for reinforcement learning training. The final column is the unweighted mean across the eight benchmark scores. Bold denotes the best result within each model-scale block; ties are both bolded.

	In-domain	**Out-of-domain**							
Method	DeepMath	AIME24	AIME25	AIME26	AMC23	MATH500	Minerva	Olympiad	Mean
				Qwen3-1.7B					
Random	**0.6496**	0.2146	0.1521	0.1375	0.5750	0.7940	0.3290	0.2642	0.3895
Balanced	0.6224	0.2167	0.1458	0.1229	0.5516	0.7900	0.3318	**0.2693**	0.3813
Classic	0.6144	0.2021	0.1146	0.1167	0.5781	0.8120	0.3336	0.2563	0.3785
Cosine	0.6024	0.1958	0.1562	**0.1396**	0.5734	0.7980	0.3364	0.2585	0.3825
Gaussian	0.6080	0.1604	0.1188	0.0938	0.5297	0.7960	0.3061	0.2491	0.3577
ODF	0.6144	0.2354	0.1750	0.1375	0.5906	0.8120	0.3382	0.2520	0.3944
CDAS	0.6144	0.1958	0.1688	0.1062	0.5500	0.7900	0.3235	0.2563	0.3756
Ours	0.6432	**0.2250**	**0.1750**	0.1208	**0.5922**	**0.8320**	**0.3401**	**0.2657**	**0.3992**
				Qwen3-4B					
Random	0.7904	0.3479	0.2833	0.3271	0.8188	0.9080	0.4430	0.3492	0.5335
Balanced	0.7856	0.3833	0.3042	**0.3604**	0.8203	0.8980	**0.4651**	0.3593	0.5470
Classic	0.7904	0.4167	0.2938	0.3479	0.8438	0.9120	0.4292	0.3521	0.5482
Cosine	**0.8024**	0.3625	0.3021	0.3354	0.8234	0.8980	0.4449	0.3456	0.5393
Gaussian	0.7880	0.3479	0.2521	0.3042	0.8250	0.9080	0.4494	0.3485	0.5279
ODF	0.7840	0.3875	0.2854	0.3396	**0.8453**	0.9180	0.4632	0.3650	0.5485
CDAS	0.7776	0.3521	0.2771	**0.3604**	0.8422	0.9000	0.4559	0.3449	0.5388
Ours	**0.8024**	**0.4250**	**0.3438**	0.3333	0.8406	**0.9200**	0.4614	**0.3672**	**0.5617**

5.4 Evaluation

We evaluate on eight mathematical reasoning benchmarks: AIME 2024, AIME 2025, AIME 2026, AMC23, DeepMath Test, MATH500, MinervaMath, and OlympiadBench. MATH500 is derived from the competition-style MATH benchmark (Hendrycks et al., 2021); MinervaMath follows the advanced quantitative-reasoning evaluation introduced with Minerva (Lewkowycz et al., 2022); and OlympiadBench targets olympiad-level mathematical and scientific reasoning (He et al., 2024). For all benchmarks, we use repeated-sampling evaluation with $k = 16$ and report avg@16 accuracy. Responses are generated with temperature 0.6 and top-p 0.95. The "Mean" column is the unweighted average across eight benchmarks.

Across both model scales, our adaptive curriculum achieves the highest aggregate performance. On Qwen3-1.7B, our method obtains a mean score of 0.3992, outperforming the strongest baseline, ODF (0.3944), by 0.0048, or 0.48 percentage points. On Qwen3-4B, our method reaches 0.5617, improving over the strongest baseline, ODF (0.5485), by 0.0135, or 1.35 percentage points.

The improvements are not caused by universal dominance on every benchmark. On Qwen3-1.7B, Random is slightly stronger on DeepMath Test, Cosine is stronger on AIME 2026, and Balanced is stronger on OlympiadBench. On Qwen3-4B, Classic is slightly stronger on AMC23, Balanced is stronger on AIME 2026 and MinervaMath, and Cosine ties our method on DeepMath Test. We therefore interpret the main result as improved aggregate robustness across heterogeneous mathematical reasoning benchmarks rather than dominance on every individual dataset.

Training trajectories. Final-checkpoint scores summarize the end result of optimization but do not show how different curricula behave during training. We therefore also track checkpoint performance throughout all 156 optimization steps. The complete eight-benchmark trajectories for Qwen3-1.7B and Qwen3-4B are shown in Appendix D (Figures 3a and 3b). The curves are intentionally used as diagnostic evidence rather than as a pointwise ranking criterion: performance is not uniformly monotonic, and different baselines may temporarily lead on individual benchmarks. The key pattern is that our competence-driven sampler remains competitive throughout training and reaches the strongest aggregate final performance at both model scales, consistent with Table 2.

Table 3: Difficulty-range ablation. Values are the unweighted mean of avg@16 accuracy across eight mathematical reasoning benchmarks. All variants use the full training sample budget while restricting the accessible difficulty range.

Method	Qwen3-1.7B	Qwen3-4B
A1: B4 only	0.3615	0.5109
A2: B3–B4	0.3510	0.5098
A3: B2–B4	0.3728	0.5534
A4: B1–B4	0.3911	0.5263
Ours: Adaptive	**0.3992**	**0.5617**

5.5 Difficulty-Range Ablation

We next ask whether a fixed subset of the difficulty range can explain the gain of our adaptive curriculum. All A1–A4 variants use the full training sample budget, while restricting the accessible difficulty range. Samples are drawn randomly from the active difficulty regions.

The ablations support two observations. First, simply emphasizing the hardest examples is not sufficient. B4-only and B3–B4 training are substantially below our adaptive curriculum at both scales. This is consistent with the role of prerequisite readiness: a large mastery gap does not imply that a region is currently the most effective training target.

Second, the strongest *fixed* difficulty range changes with model scale. For Qwen3-1.7B, full-range A4 is the strongest fixed-range variant at 0.3911, suggesting that the smaller model benefits from retaining broad difficulty coverage. In contrast, for Qwen3-4B, A3 (B2–B4) reaches 0.5534 and clearly outperforms full-range A4 (0.5263), indicating that the stronger model benefits more from concentrating on intermediate and difficult regions.

Our adaptive curriculum remains best in both cases, reaching 0.3992 at 1.7B and 0.5617 at 4B. This result supports the central hypothesis that there is no universally optimal fixed difficulty support: the appropriate training distribution depends on the learner's competence and should therefore adapt online.

Trajectory-level evidence. The final-checkpoint comparison is complemented by the full optimization trajectories in Appendix D (Figures 4a and 4b). For Qwen3-1.7B, the full-range A4 variant remains the strongest fixed-range competitor over much of training, consistent with its final aggregate score. For Qwen3-4B, A3 (B2–B4) is the strongest fixed-range competitor, whereas hard-only A1/A2 remain less competitive overall. The trajectories therefore support the same cross-scale conclusion as Table 3: the preferred fixed difficulty support changes with model capacity, while the competence-driven curriculum adapts without selecting that support in advance. We additionally examine the corresponding training-batch reward and response-length dynamics in Figures 5a and 5b. These diagnostics help distinguish curriculum effects from a trivial increase in generated response length.

5.6 Curriculum Dynamics

The learned quotas provide direct evidence that the scheduler implements competence-driven progression. For a representative Qwen3-4B run, calibration and the resulting scheduler state are approximately

$$\bar{\mathbf{p}}_0 = [0.512,\, 0.475,\, 0.369,\, 0.221], \qquad \mathbf{R}_0 = [1.000,\, 0.512,\, 0.243,\, 0.090], \tag{17}$$

with the batch allocation evolving from

$$\mathbf{q}_{\text{early}} \approx [31, 18, 10, 5] \quad \longrightarrow \quad \mathbf{q}_{\text{late}} \approx [22, 17, 14, 11]. \tag{18}$$

Thus B1 decreases from roughly 31 to 22 prompts per batch, whereas B4 increases from 5 to 11. This transition is not specified by a step-based schedule; it emerges from online competence estimates.

Full cross-scale quota trajectories are provided immediately after Algorithm 1 in Appendix A (Figures **??** and **??**). These trajectories expose the closed loop behind the aggregate quota shift: changes in online competence alter readiness and mastery signals, which in turn redistribute the next training batch across difficulty regions.

6 DISCUSSION

Our results support a simple view of curriculum sampling for RLVR: the most useful difficulty region is neither necessarily the easiest nor the hardest, but one for which the model has acquired sufficient prerequisite ability while still retaining substantial room for improvement.

The difficulty-range ablations are informative. Hard-only training performs poorly at both model scales, indicating that the benefit of our method cannot be explained by merely exposing the model to more difficult data. At the same time, the strongest fixed range changes from B1–B4 for Qwen3-1.7B to B2–B4 for Qwen3-4B. This model-scale dependence motivates a curriculum that responds to competence rather than using one fixed support for all learners.

Cumulative readiness is intentionally conservative. For the hardest region,

$$R_4 = \bar{p}_1 \bar{p}_2 \bar{p}_3. \tag{19}$$

This suppresses premature sampling of very hard examples, but the multiplicative structure may also attenuate the readiness of deep buckets when any prerequisite competence remains moderate. In our observed 4B trajectory, B4 nevertheless increases from approximately 5 to 11 prompts per batch, showing that the curriculum frontier moves as prerequisite competence improves.

More broadly, our method differs from the closest adaptive sampling strategies in both granularity and control signal. E2H Reasoner uses training-progress schedules such as cosine and Gaussian sampling (Parashar et al., 2026); ODF emphasizes samples with intermediate current success rates (Bae et al., 2026); and CDAS estimates historical problem difficulty and aligns it with a global competence scalar (Kong et al., 2025). Our scheduler instead operates over ordered difficulty regions and uses the full easier-bucket chain to estimate prerequisite readiness, while the target bucket's mastery gap supplies the remaining learning need. Compared with per-problem adaptive methods such as CDAS, our scheduler maintains only $O(K)$ bucket-level states rather than tracking individual examples, and directly controls the training-budget allocation.

7 LIMITATIONS

This study has two limitations. First, our method relies on predefined DeepMath difficulty labels and assumes an ordered prerequisite structure among buckets. Extending adaptive curriculum allocation to learned or partially ordered difficulty structures is an important direction. Second, the bucket-level adaptation improves efficiency and stability but may overlook instance-level difficulty variations within each bucket.

8 CONCLUSION

We introduced a competence-driven curriculum sampler for reinforcement learning with verifiable rewards. The method maintains online bucket-level competence estimates and decomposes sampling priority into cumulative prerequisite readiness and remaining mastery gap:

$$s_{k,t} = \left(\prod_{j<k} \bar{p}_{j,t-1} \right) \left(1 - \bar{p}_{k,t-1} + \epsilon\right). \tag{20}$$

This design operationalizes the principle of training difficulty regions that are *ready to learn but not yet mastered*. Curriculum progression is induced directly by the model's evolving competence rather than by a predefined training-time schedule.

Across both Qwen3-1.7B and Qwen3-4B, our experiments achieve the highest eight-benchmark mean among the compared sampling baselines. Moreover, the strongest fixed difficulty range differs across model scales, whereas our adaptive scheduler remains best at both scales. These findings suggest that curriculum allocation should be competence-dependent rather than fixed in advance.

REPRODUCIBILITY STATEMENT

We provide the difficulty-bucket construction, calibration procedure, competence update, readiness and mastery equations, quota allocation rule, training configuration, and evaluation protocol in the main paper and appendix. The adaptive scheduler modifies only data sampling and leaves the GRPO reward, advantage computation, loss, and optimizer unchanged. We record competence estimates, readiness values, scores, and per-bucket quotas throughout training. Anonymized source code, launch scripts, and exact run configurations will be included in the supplementary material.

Ethics Statement

This work studies optimization and data-sampling strategies for mathematical reasoning models and does not involve human subjects or private user data. The main broader consideration is that improvements in general reasoning systems may transfer to downstream applications beyond mathematics. We therefore limit our empirical claims to mathematical reasoning benchmarks and do not advocate deployment in high-stakes settings.

AI Use Statement

Generative AI tools were used to assist with code implementation and debugging, as well as language polishing and editing. The research ideas, methodology, experiments, and conclusions were developed and verified by the authors.

References

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, Jeongyeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, 2026. doi: 10.18653/v1/2026.eacl-long.30.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Xinjie Chen, Minpeng Liao, Guoxin Chen, Chengxi Li, Biao Fu, Kai Fan, and Xinggao Liu. From data-centric to sample-centric: Enhancing LLM reasoning via progressive optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2227–2242, 2026. doi: 10.18653/v1/2026.acl-long.101.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024. doi: 10.18653/v1/2024.acl-long.211.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

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

Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. What makes a good curriculum? disentangling the effects of data ordering on LLM mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34472–34488, 2026. doi: 10.18653/v1/2026.acl-long.1591.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2304–2313, 2018.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1613.

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. *arXiv preprint arXiv:2505.17652*, 2025.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. In *International Conference on Learning Representations*, 2026.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1162–1172, 2019. doi: 10.18653/v1/N19-1119.

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

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

Zelin Tan, Hejia Geng, Xiaohang Yu, Mulei Zhang, Guancheng Wan, Yifan Zhou, Qiang He, Xiangyuan Xue, Heng Zhou, Yutao Fan, Zhong-Zhi Li, Zaibin Zhang, Guibin Zhang, Chen Zhang, Zhenfei Yin, Philip Torr, and Lei Bai. Scaling behaviors of LLM reinforcement learning post-training: An empirical study in mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31300–31319, 2026. doi: 10.18653/v1/2026.acl-long.1444.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1800.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3775.

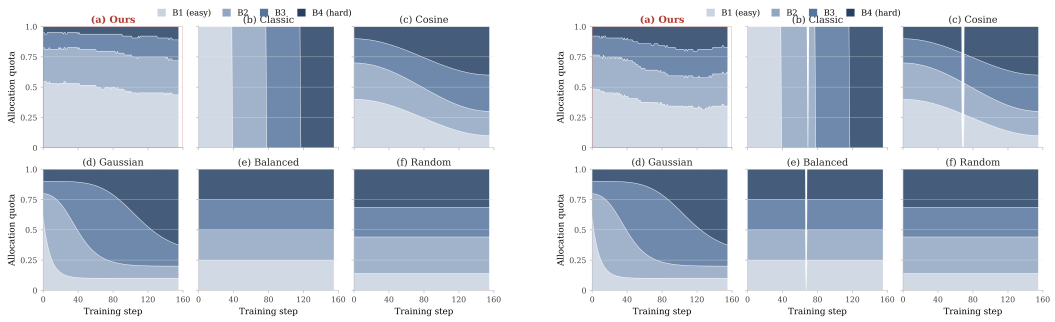
Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond random sampling: Efficient language model pretraining via curriculum learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 5776–5794, 2026. doi: 10.18653/v1/2026.eacl-long.271.

Zirui Zhao, Hanze Dong, Amrita Saha, Caiming Xiong, and Doyen Sahoo. Automatic curriculum expert iteration for reliable LLM reasoning. In *International Conference on Learning Representations*, 2025.

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. Data efficient RLVR via off-policy influence guidance. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46167–46192, 2026. doi: 10.18653/v1/2026.acl-long.2141.

A Complete Training Procedure

Algorithm 1 gives the complete training procedure. The scheduler is initialized by bucket-wise no-gradient calibration and then operates in a closed loop: the current competence state determines readiness, mastery gap, and per-bucket quotas; GRPO updates the policy on the resulting batch; and the new on-policy outcomes update the competence estimates used by the next curriculum decision. Algorithm 1 converts the online competence state into a per-bucket sampling quota at every training step. We therefore place the realized quota trajectories directly after the complete algorithm, so that the reader can inspect how the proposed closed loop changes the actual training distribution before moving to the remaining experimental diagnostics.



(a) Qwen3-1.7B (b) Qwen3-4B

Figure 2: **Per-bucket quota dynamics of the adaptive curriculum scheduler.** The scheduler dynamically reallocates training samples across ordered difficulty buckets according to cumulative readiness and mastery gap. Easier regions dominate early training, while sampling gradually shifts toward harder regions as prerequisite competence improves.

Interpretation. The quota trajectories provide direct mechanism-level evidence for the readiness–mastery design. Prerequisite improvement raises the accessibility of harder regions, while increasing mastery suppresses the priority of regions that need less additional training. The curriculum frontier therefore moves with the learner state instead of following a fixed easy-to-hard clock.

B DeepMath-103K Data Construction and Split Protocol

This section describes the construction of the DeepMath training, validation, and test subsets used throughout the experiments. The split is generated once with a fixed random seed and is reused across all methods and model scales.

B.1 Source Data and Preprocessing

The source corpus consists of 103,022 examples from DeepMath-103K, stored in ten Parquet shards. Each example provides the fields `question`, `final_answer`, `topic`, and `difficulty`. The `topic` field is hierarchical (e.g., `Mathematics -> Algebra -> ...`), and `difficulty` is a floating-point difficulty score.
We apply the following deterministic preprocessing steps:

1. Remove examples with empty required fields. Two examples with empty `final_answer` are removed, leaving 103,020 examples.
2. Deduplicate examples by a normalized version of the question text. This removes 1,140 duplicate questions and leaves 101,880 unique examples.

Algorithm 1 Competence-Driven Curriculum Sampling for RLVR

Require: Ordered difficulty buckets $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$; initial policy π_θ; verifiable reward function $r(\cdot)$; rollout group size G; global prompt batch size B; EMA coefficient α; score offset ϵ; training steps T

Ensure: Optimized policy π_θ

Calibration
for $k = 1, \ldots, K$ **do**
  Sample calibration prompts $\mathcal{C}_k \subset \mathcal{B}_k$
  Generate no-gradient rollouts with π_θ on $\mathcal{C}_k$
  Compute calibration success rate p_k^{cal}
  Initialize $\bar{p}_{k,0} \leftarrow p_k^{\text{cal}}$
end for

Curriculum-guided RL training
for $t = 1, \ldots, T$ **do**
  $R_{1,t} \leftarrow 1$
  for $k = 2, \ldots, K$ **do**
    $R_{k,t} \leftarrow \prod_{j<k} \bar{p}_{j,t-1}$
  end for
  for $k = 1, \ldots, K$ **do**
    $M_{k,t} \leftarrow 1 - \bar{p}_{k,t-1}$
    $s_{k,t} \leftarrow R_{k,t}\big(M_{k,t} + \epsilon\big)$
  end for
  $\{q_{k,t}\}_{k=1}^{K} \leftarrow \text{LargestRemainderAllocation}(\{s_{k,t}\}_{k=1}^{K}, B)$ subject to $\sum_{k=1}^{K} q_{k,t} = B$ and $q_{k,t} \geq 1$
  Construct $\mathcal{D}_t$ by sampling $q_{k,t}$ prompts from each $\mathcal{B}_k$
  for all $x \in \mathcal{D}_t$ **do**
    Sample G on-policy responses $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$
    Compute verifiable rewards $\{r_i\}_{i=1}^{G}$
    Compute GRPO advantages from the rollout group
  end for
  Update π_θ using the standard GRPO objective
  for $k = 1, \ldots, K$ **do**
    if $\mathcal{D}_t \cap \mathcal{B}_k \neq \varnothing$ **then**
      Compute online bucket success rate $p_{k,t}$
      $\bar{p}_{k,t} \leftarrow (1 - \alpha)\bar{p}_{k,t-1} + \alpha p_{k,t}$
    else
      $\bar{p}_{k,t} \leftarrow \bar{p}_{k,t-1}$
    end if
  end for
end for
return π_θ

3. Preserve the original hierarchical topic as `topic_raw`. Because the first hierarchy level is `Mathematics` for all examples, we define `topic_norm` using the second hierarchy level, resulting in nine topic categories.
4. Treat `difficulty` $= -1$ as an unknown-value sentinel. Only four examples have this value; they are assigned to an `UNK` difficulty group during preprocessing.

The resulting nine normalized topic categories are Algebra, Calculus, Precalculus, Geometry, Applied Mathematics, Discrete Mathematics, Number Theory, Other, and Differential Equations.

Table 4: Final DeepMath split sizes after cleaning, question-level deduplication, and joint topic–difficulty stratification.

Stage / split	Number of examples
Original DeepMath-103K	103,022
After required-field cleaning	103,020
After question deduplication	101,880
Train	10,000
Validation	1,250
Test	1,250

Table 5: Normalized topic distribution of the DeepMath train, validation, and test subsets. Parentheses show proportions for the two largest topics; the remaining counts follow the same stratification procedure.

Topic	Train	Validation	Test
Algebra	3,188 (0.3188)	399 (0.3192)	399 (0.3192)
Calculus	2,733 (0.2733)	342 (0.2736)	341 (0.2728)
Precalculus	862	108	109
Geometry	824	102	102
Applied Mathematics	768	96	96
Discrete Mathematics	569	71	70
Number Theory	563	70	71
Other	339	42	43
Differential Equations	154	20	19

B.2 Joint Topic–Difficulty Stratification

We construct the final subsets by stratifying jointly over normalized topic and difficulty. The difficulty bins are

$$\text{D3_4} : d \leq 4, \tag{21}$$
$$\text{D5} : d = 5, \tag{22}$$
$$\text{D6} : d = 6, \tag{23}$$
$$\text{D7} : d = 7, \tag{24}$$
$$\text{D8_9} : d \geq 8. \tag{25}$$

Combining nine topic categories with five non-empty difficulty bins yields 45 topic–difficulty strata. All 45 strata that appear in the training split also appear in both validation and test splits. The four `UNK` difficulty examples account for approximately 0.004% of the deduplicated corpus and receive zero allocation under the target split sizes.
We use the fixed seed

$$\texttt{seed} = 20260707 \tag{26}$$

and sample

$$10{,}000/1{,}250/1{,}250 \tag{27}$$

examples for train/validation/test, respectively. The three subsets are strictly disjoint by UID, and each subset contains unique UIDs.

B.3 Topic Distribution

Table 5 reports the normalized topic distribution of the three subsets. The topic proportions are closely matched across splits; the maximum absolute cross-split proportion difference is 0.0010.

B.4 Difficulty Distribution

Table 6 reports the difficulty-bin distribution. The maximum absolute cross-split proportion difference is 0.0023, indicating that the train, validation, and test subsets preserve nearly identical difficulty mixtures.
The four curriculum buckets used by our sampler are obtained from these difficulty bins as

$$B_1 = \text{D3_4}, \quad B_2 = \text{D5}, \quad B_3 = \text{D6}, \quad B_4 = \text{D7} \cup \text{D8_9}. \tag{28}$$

Table 6: Difficulty-bin distribution of the DeepMath train, validation, and test subsets.

Difficulty bin	Train	Validation	Test
D3_4	1,393	173	173
D5	3,016	377	377
D6	2,443	305	304
D7	1,483	186	185
D8_9	1,665	209	211

B.5 Integrity and Leakage Checks

We perform several integrity checks before training. First, train, validation, and test UID sets are pairwise disjoint, and every split contains unique UIDs. Second, all required fields are present after preprocessing. Third, the generated training prompt is verified structurally to equal the fixed prompt prefix plus the original question plus the fixed suffix. The prompt contains an empty answer placeholder only and does not include the ground-truth `final_answer` or any retained solution field. This structural check avoids false positive "answer leakage" detections caused by short answers such as single digits or letters appearing naturally as substrings in the problem statement.
The final training records retain the prompt and ground-truth answer required for GRPO. Additional solution fields are preserved only for possible future supervised fine-tuning experiments and are not inserted into the RLVR prompt.

B.6 Reproducibility

The split-generation script, fixed seed, input-shard list, field mapping, cleaning statistics, selected UIDs, and per-topic/per-difficulty summary statistics are stored with the processed dataset. The original DeepMath data are not modified. These artifacts allow the split to be reproduced exactly from the original shards.

C Additional Experimental Results

Difficulty-range ablation protocol. For A1–A4, all variants use the full training sample budget. The only difference is the set of difficulty regions available for random sampling: A1 uses B4, A2 uses B3–B4, A3 uses B2–B4, and A4 uses the full B1–B4 range. A4 differs from the Balanced baseline because it does not enforce equal per-bucket counts within each batch.

C.1 Full Difficulty-Range Ablation on Qwen3-1.7B

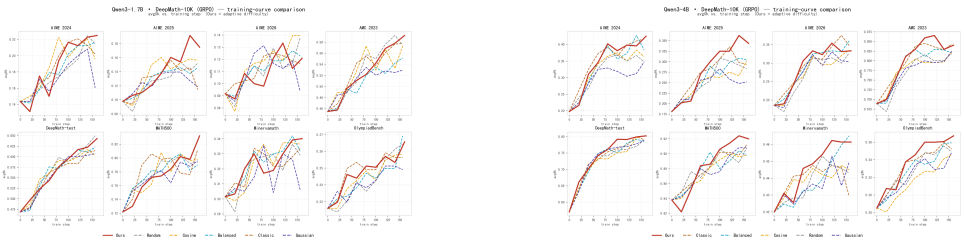
Table 7: Full difficulty-range ablation on Qwen3-1.7B. All variants use the full training sample budget while restricting the accessible difficulty range.

Method	AIME24	AIME25	AIME26	AMC23	DeepMath	MATH500	Minerva	Olympiad	Mean
Ours	0.2250	0.1750	0.1208	0.5922	0.6432	0.8320	0.3401	0.2657	**0.3992**
A1: B4 only	0.1854	0.1208	0.1271	0.5125	0.6016	0.7700	0.3336	0.2412	0.3615
A2: B3–B4	0.1479	0.1146	0.0875	0.5312	0.6064	0.7580	0.3272	0.2354	0.3510
A3: B2–B4	0.2167	0.1292	0.1021	0.5328	0.6232	0.8040	0.3244	0.2498	0.3728
A4: B1–B4	0.2354	0.1417	0.1583	0.5859	0.6136	0.8040	0.3336	0.2563	0.3911

C.2 Full Difficulty-Range Ablation on Qwen3-4B

Table 8: Full difficulty-range ablation on Qwen3-4B. All variants use the full training sample budget while restricting the accessible difficulty range.

Method	AIME24	AIME25	AIME26	AMC23	DeepMath	MATH500	Minerva	Olympiad	Mean
Ours	0.4250	0.3438	0.3333	0.8406	0.8024	0.9200	0.4614	0.3672	**0.5617**
A1: B4 only	0.3417	0.2479	0.2646	0.7922	0.7656	0.8900	0.4550	0.3305	0.5109
A2: B3–B4	0.3292	0.2521	0.2729	0.7797	0.7816	0.8840	0.4449	0.3341	0.5098
A3: B2–B4	0.3958	0.3312	0.3354	0.8234	0.8008	0.9140	0.4678	0.3585	0.5534
A4: B1–B4	0.3083	0.2812	0.3188	0.8328	0.7776	0.9060	0.4559	0.3297	0.5263



(a) Qwen3-1.7B (b) Qwen3-4B

Figure 3: **Evaluation trajectories during RLVR training across model scales.** Each panel reports checkpoint performance on eight mathematical reasoning benchmarks over 156 optimization steps. We compare our competence-driven curriculum with Random, Balanced, Classic, Cosine (E2H-C), and Gaussian (E2H-G) sampling. These trajectory plots complement the final-checkpoint results in Table 2 by showing how performance evolves throughout training on Qwen3-1.7B and Qwen3-4B. While no single method dominates every benchmark at every checkpoint, our method achieves the strongest aggregate final performance across scales.

D Training Curves, Scheduler Dynamics, and Optimization Diagnostics

To complement the final-checkpoint results, we report trajectory-level diagnostics over the complete 156-step RLVR run. We separate two comparisons: (1) our method against the standard sampling baselines in Table 2, and (2) our method against the fixed difficulty-range ablations in Table 3. Each panel uses the same benchmark metric as the main evaluation and plots checkpoint performance against training step. We use these curves to characterize optimization dynamics rather than to claim pointwise dominance at every checkpoint.

D.1 Standard Sampling Baselines

Interpretation. The two model scales exhibit visibly different optimization trajectories, which is consistent with the difficulty-range ablation: the curriculum that is most useful depends on the learner state and model capacity. In particular, temporary benchmark-specific advantages of static schedules do not translate into a universally best fixed curriculum. Our competence-driven sampler instead updates its training allocation from online bucket competence, which supports stronger aggregate performance at the end of training across both model scales.

D.2 Difficulty-Range Ablation Trajectories

Figures 4a and 4b show how the fixed-support ablations evolve during training. Unlike the standard-baseline comparison, these experiments vary only the set of difficulty regions available to the learner: A1 uses B4, A2 uses B3–B4, A3 uses B2–B4, and A4 uses B1–B4.

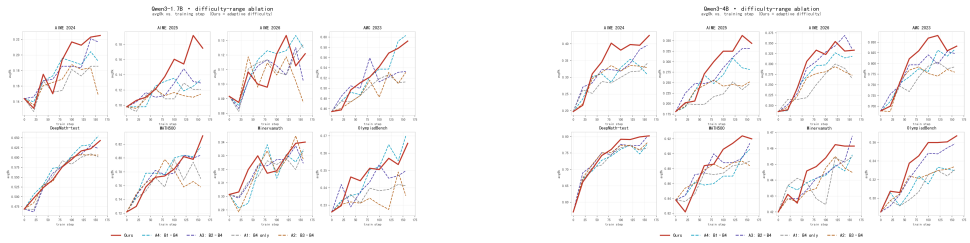
D.3 Reward and Response-Length Dynamics

Benchmark trajectories characterize downstream performance, while the training-batch reward and generated response length provide a complementary view of the optimization process. Figures 5a and 5b report these two diagnostics for the difficulty-range ablations. The upper panel shows the mean verifiable reward of the current training batch, and the lower panel shows the mean response length produced during rollout.

D.4 Adaptive-Run Reward and Response-Length Diagnostics

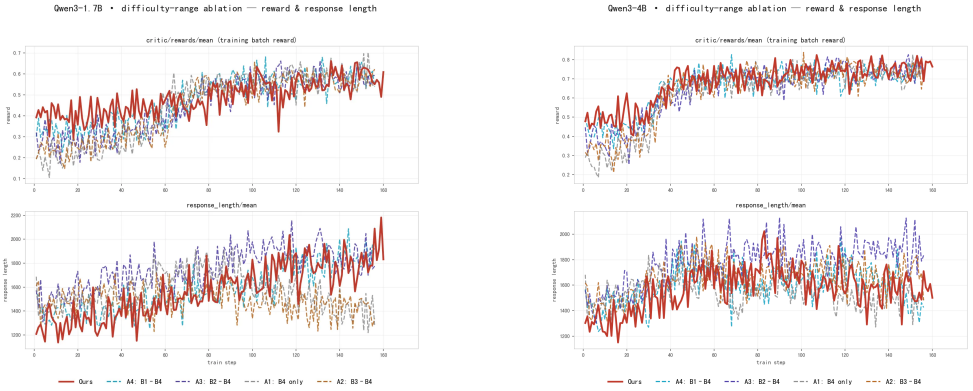
The preceding A1–A4 comparison examines reward and response-length behavior across fixed difficulty supports. For completeness, Figures 6a and 6b show the corresponding reward and response-length trajectories recorded for the adaptive curriculum runs themselves.

Optimization-level interpretation. Across both model scales, curriculum choice changes not only downstream benchmark trajectories but also the path taken by the policy during RLVR. The reward curves generally rise as training proceeds, whereas response length evolves differently across fixed difficulty supports and model scales. Most importantly, stronger final benchmark performance does not correspond to a universal "longer is better" pattern. This is consistent with our interpreta-



(a) Qwen3-1.7B (b) Qwen3-4B

Figure 4: **Difficulty-range ablation trajectories across model scales.** Each panel reports checkpoint performance on eight mathematical reasoning benchmarks over 156 optimization steps. We compare our competence-driven curriculum with four fixed-range variants: A1 (B4 only), A2 (B3–B4), A3 (B2–B4), and A4 (B1–B4). The two model scales exhibit different preferences over fixed difficulty supports: A4 is the strongest fixed-range variant at the final checkpoint on Qwen3-1.7B, whereas A3 is the strongest fixed-range variant on Qwen3-4B. In both cases, however, the proposed adaptive sampler achieves the strongest aggregate final performance, showing that it can match or exceed the best fixed difficulty support without requiring manual range selection.



(a) Qwen3-1.7B (b) Qwen3-4B

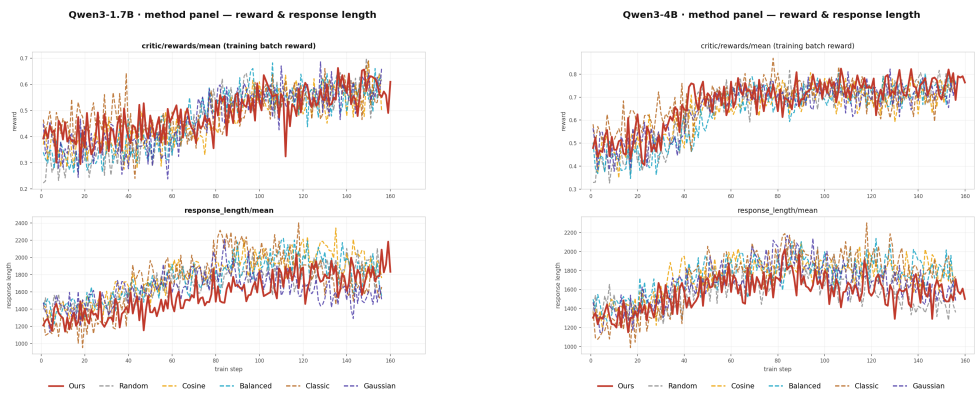
Figure 5: **Training-batch reward and response-length dynamics for difficulty-range ablations across model scales.** The upper panels show mean training-batch reward and the lower panels show mean rollout response length over 156 optimization steps. We compare our competence-driven curriculum with four fixed-support variants: A1 (B4 only), A2 (B3–B4), A3 (B2–B4), and A4 (B1–B4). Across both model scales, reward generally improves for all variants as training progresses, while the response-length trajectories differ substantially across difficulty supports. This indicates that the performance advantage of the adaptive curriculum is not explained simply by producing uniformly longer responses.

tion that the sampler improves *which difficulty regions are used for learning*, rather than obtaining gains merely by inducing longer reasoning traces.

Cross-scale interpretation. The trajectory comparison reinforces the key ablation result without requiring any single method to dominate every benchmark at every checkpoint. The smaller Qwen3-1.7B model benefits most from retaining the full B1–B4 support among fixed-range variants, whereas Qwen3-4B benefits more from the B2–B4 range. In contrast, the same competence-driven scheduler is used at both scales and adapts the per-bucket allocation online. This provides trajectory-level support for our claim that the useful difficulty frontier is learner-dependent rather than fixed a priori.

E Theoretical Analysis of the Curriculum Scheduler

This section establishes structural properties of the curriculum scheduler. These results characterize the scheduler itself and do not constitute a global convergence guarantee for GRPO, whose dynamics



(a) Qwen3-1.7B (b) Qwen3-4B

Figure 6: **Reward and response-length diagnostics for the adaptive curriculum across model scales.** The upper panels report mean training-batch reward and the lower panels report mean rollout response length over the 156-step optimization trajectory. These diagnostics provide additional views of the optimization dynamics of the proposed curriculum on Qwen3-1.7B and Qwen3-4B.

involve non-convex neural optimization, on-policy distribution shift, stochastic rewards, and cross-bucket transfer.

Recall

$$R_1 = 1, \qquad R_k = \prod_{j<k} \bar{p}_j, \tag{29}$$

$$M_k = 1 - \bar{p}_k, \tag{30}$$

and

$$s_k = R_k(M_k + \epsilon) = \left(\prod_{j<k} \bar{p}_j \right) (1 - \bar{p}_k + \epsilon), \tag{31}$$

where $\bar{p}_k \in [0, 1]$ and $\epsilon \geq 0$.

E.1 Prerequisite Monotonicity

Proposition 3. For any prerequisite bucket $j < k$,

$$\frac{\partial s_k}{\partial \bar{p}_j} \geq 0. \tag{32}$$

Proof. From Eq. 31,

$$\frac{\partial s_k}{\partial \bar{p}_j} = \left(\prod_{\substack{\ell<k \\ \ell \neq j}} \bar{p}_\ell \right) (1 - \bar{p}_k + \epsilon). \tag{33}$$

All factors are non-negative because $\bar{p}_\ell \in [0, 1]$ and $\epsilon \geq 0$. Hence the derivative is non-negative. □

E.2 Mastery Suppression

Proposition 4. For the target bucket,

$$\frac{\partial s_k}{\partial \bar{p}_k} = -R_k \leq 0. \tag{34}$$

Proof. R_k depends only on prerequisite buckets $j < k$, so

$$\frac{\partial}{\partial \bar{p}_k} R_k(1 - \bar{p}_k + \epsilon) = -R_k. \tag{35}$$

Since $R_k \geq 0$, the derivative is non-positive. □

E.3 Suppression of Unready and Mastered Regions

Proposition 5. In the idealized case $\epsilon = 0$:

1. if any prerequisite competence $\bar{p}_j = 0$ for $j < k$, then $s_k = 0$;
2. if the target bucket is fully mastered, $\bar{p}_k = 1$, then $s_k = 0$.

Proof. If a prerequisite is zero, then $R_k = \prod_{j<k} \bar{p}_j = 0$, hence $s_k = 0$. If $\bar{p}_k = 1$, then $M_k = 1 - \bar{p}_k = 0$, hence $s_k = R_k M_k = 0$. □

This property formalizes the two-sided suppression intended by the design: currently unready regions and already-mastered regions both receive low dominant priority. In the implementation, $\epsilon > 0$ and $q_{\min} \geq 1$ preserve exploration.

E.4 Endogenous Promotion Boundary

For adjacent buckets,

$$s_k = R_k(1 - \bar{p}_k + \epsilon), \tag{36}$$

$$s_{k+1} = R_k \bar{p}_k (1 - \bar{p}_{k+1} + \epsilon). \tag{37}$$

Proposition 6. Assuming $R_k > 0$, bucket $k+1$ has a larger score than bucket k if and only if

$$\boxed{\bar{p}_k > \frac{1+\epsilon}{2 - \bar{p}_{k+1} + \epsilon}.} \tag{38}$$

Proof. Starting from $s_{k+1} > s_k$ and canceling $R_k > 0$ gives

$$\bar{p}_k(1 - \bar{p}_{k+1} + \epsilon) > 1 - \bar{p}_k + \epsilon. \tag{39}$$

Rearranging,

$$\bar{p}_k(2 - \bar{p}_{k+1} + \epsilon) > 1 + \epsilon, \tag{40}$$

which yields Eq. 38. □

The cumulative-readiness score induces the following state-dependent promotion boundary

$$\beta^{\star}_{k \to k+1} = \frac{1+\epsilon}{2 - \bar{p}_{k+1} + \epsilon}. \tag{41}$$

Thus transitions between adjacent difficulty regions emerge endogenously from the current competence state.

E.5 Why the Curriculum Frontier Moves

For $s_k > 0$,

$$\log s_k = \sum_{j<k} \log \bar{p}_j + \log(1 - \bar{p}_k + \epsilon). \tag{42}$$

Differentiating with respect to training time gives

$$\frac{d}{dt} \log s_k = \sum_{j<k} \frac{\dot{\bar{p}}_j}{\bar{p}_j} - \frac{\dot{\bar{p}}_k}{1 - \bar{p}_k + \epsilon}. \tag{43}$$

The first term increases a harder bucket's score as its prerequisites improve, while the second decreases its score as the target bucket itself becomes mastered. This provides a simple explanation for a moving curriculum frontier: a bucket gains priority while prerequisites are being acquired and loses priority after its own mastery improves.

E.6 Optimization View of Quota Allocation

Let every non-empty bucket first receive $q_{\min}$ samples. The remaining continuous budget is

$$C = B - K q_{\min}. \tag{44}$$

Let $x_k > 0$ denote additional quota with $\sum_k x_k = C$. Consider

$$\max_{\{x_k\}} \sum_{k=1}^{K} s_k \log x_k. \tag{45}$$

Proposition 7. For positive s_k, the unique optimum is

$$\boxed{x_k^\star = C\frac{s_k}{\sum_j s_j}.} \tag{46}$$

Proof. The Lagrangian is

$$\mathcal{L} = \sum_k s_k \log x_k - \lambda\left(\sum_k x_k - C\right). \tag{47}$$

The first-order condition gives

$$\frac{s_k}{x_k} - \lambda = 0 \quad \Rightarrow \quad x_k = \frac{s_k}{\lambda}. \tag{48}$$

Applying the budget constraint yields

$$\lambda = \frac{\sum_j s_j}{C}, \tag{49}$$

and therefore

$$x_k^\star = C\frac{s_k}{\sum_j s_j}. \tag{50}$$

Since $\log x$ is strictly concave, the objective is strictly concave for positive weights, so the stationary point is the unique global optimum. □

The largest-remainder rule used in implementation is an integer realization of this continuous proportional allocation.

E.7 EMA Competence Tracking

Suppose that over a short interval the true bucket success probability is locally constant at p and the observed batch success rate is

$$\hat{p}_t = p + \xi_t, \qquad \mathbb{E}[\xi_t] = 0, \qquad \mathrm{Var}(\xi_t) = \sigma^2. \tag{51}$$

The EMA update is

$$\bar{p}_t = (1-\alpha)\bar{p}_{t-1} + \alpha\hat{p}_t. \tag{52}$$

Proposition 8 (EMA mean convergence). Under the local stationarity assumption,

$$\mathbb{E}[\bar{p}_t] - p = (1-\alpha)^t(\bar{p}_0 - p), \tag{53}$$

and therefore $\mathbb{E}[\bar{p}_t] \to p$.

Proof. Taking expectations and subtracting p gives

$$\mathbb{E}[\bar{p}_t] - p = (1-\alpha)\left(\mathbb{E}[\bar{p}_{t-1}] - p\right). \tag{54}$$

Recursion yields the result. □

Proposition 9 (EMA variance). If observation noise is independent across updates, the stationary EMA variance is

$$\boxed{\mathrm{Var}(\bar{p}) = \frac{\alpha}{2-\alpha}\sigma^2.} \tag{55}$$

Proof. At stationarity let $V = \mathrm{Var}(\bar{p}_t)$. Independence gives

$$V = (1-\alpha)^2 V + \alpha^2\sigma^2. \tag{56}$$

Thus

$$V[1-(1-\alpha)^2] = \alpha^2\sigma^2, \tag{57}$$

and since $1-(1-\alpha)^2 = \alpha(2-\alpha)$,

$$V = \frac{\alpha}{2-\alpha}\sigma^2. \tag{58}$$

□

These results expose the standard stability–responsiveness trade-off: smaller α suppresses rollout noise more strongly, whereas larger α tracks policy changes more quickly.

E.8 Probabilistic Interpretation

Let $Z_j \in \{0, 1\}$ denote whether prerequisite ability j is mastered and suppose

$$P(Z_j = 1) \approx \bar{p}_j. \tag{59}$$

Under a simplified independence approximation,

$$P(Z_1 = 1, \ldots, Z_{k-1} = 1) \approx \prod_{j<k} \bar{p}_j = R_k. \tag{60}$$

Similarly,

$$P(Z_k = 0) \approx 1 - \bar{p}_k = M_k. \tag{61}$$

Ignoring the exploration floor, $s_k = R_k M_k$ can therefore be interpreted as a proxy for the event

$$\boxed{\text{prerequisites mastered} \quad \text{and} \quad \text{target not yet mastered.}} \tag{62}$$

The empirical success rates are not exact latent mastery probabilities and the bucket events need not be independent, so this is a motivation for the functional form rather than an exact probabilistic model.

F Training Details

Prompt template. We use a rule-based verifiable-reward setup for mathematical reasoning. The prompt template used by the training pipeline is summarized below.

Table 9: Training prompt used in the RLVR experiments.

Training prompt

`Question: {question}`
`Please reason step by step and put your final answer in the required format.`
`Answer:`

Curriculum-baseline provenance. The Classic, Balanced, Cosine (E2H-C), and Gaussian (E2H-G) baselines follow the schedule family introduced by E2H Reasoner (Parashar et al., 2026). In particular, E2H-G uses a moving Gaussian center over ordered difficulty levels, whereas E2H-C uses a cosine transition from easier to harder regions. Our method does not use training progress as the curriculum state; its quotas are recomputed from online bucket competence.

Common training setup. All reported runs are trained for 156 optimization steps. The reported runs use GRPO (Shao et al., 2024) with 64 prompts per global batch and 8 rollout responses per prompt. Rollouts use vLLM (Kwon et al., 2023), maximum response length 4096, temperature 1.0, top-p 1.0, and top-$k = -1$. We use AdamW (Loshchilov & Hutter, 2019) with constant learning rate 10^{-6}, policy mini-batch size 32, actor micro-batch size 4, and clipping ratio 0.2. No explicit KL penalty is added to the reward or actor loss. Training is performed on a single node with 8 GPUs using FSDP2/full sharding and gradient checkpointing.

Evaluation setup. During evaluation, we use temperature 0.6 and top-p 0.95. All eight benchmarks are evaluated with repeated sampling using $k = 16$. For each problem, we independently generate 16 responses and compute avg@16 as the average correctness over these responses. Accordingly, all benchmark scores reported in the main result tables are avg@16 accuracies. The reported "Mean" is the unweighted average of the avg@16 scores across AIME 2024, AIME 2025, AIME 2026, AMC23, DeepMath Test, MATH500, MinervaMath, and OlympiadBench.