Unlocking the Unsolvable: Teacher-Guided Curriculum for Data-Efficient RLVR

Yukang Zhu
Independent Researcher
zhuyukang2000@gmail.com

Zhen Han*
Amazon
zhenhz@amazon.com

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has shown remarkable success in improving the mathematical reasoning of large language models. Yet problems beyond the model's current capability, where rollouts uniformly fail and no learning signal is produced, are structurally wasted despite marking the most informative training frontier. We show that these otherwise-inert problems can be *unlocked* via teacher-guided curriculum learning: partial reasoning traces from a stronger model create a graded difficulty landscape, and a backward-chaining curriculum progressively withdraws guidance until the model solves problems unaided. Training on only 128 unsolvable problems matches or exceeds GRPO trained on a full 2,000-problem corpus ($\sim$16$\times$ data efficiency) on the nine-benchmark average for both base models, while substantially expanding the reasoning boundary measured by pass@k at large k. Furthermore, we identify a distribution-shift cost that is particularly acute in the unsolvable-only regime and propose **Monotone Frontier Curriculum (MFC)**, a method that monotonically drives training toward unguided solving, consistently outperforming existing curriculum methods.

1 Introduction

Reasoning-centric large language models (LLMs) (OpenAI, 2024; Guo et al., 2025) have established Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024) as a powerful recipe for improving mathematical reasoning. RLVR builds on the broader paradigm of reinforcement learning from human feedback (Ouyang et al., 2022) and policy gradient methods (Schulman et al., 2017), replacing costly human annotations with verifiable reward signals. While most work focuses on algorithmic refinements (Yu et al., 2025; Liu et al., 2025b; Cui et al., 2025), a complementary question has received growing attention: *which training data matters most*? Pikus et al. (2025) show that the hardest problems yield the largest gains, and Foster et al. (2025) show that expected policy improvement is bounded by the variance of success. Wang et al. (2025) demonstrate that even a single example can unlock substantial reasoning gains. These findings suggest a common theme: problems near the model's reliable-solving boundary, with mixed rollout outcomes, tend to provide the richest learning signal for RLVR.

A separate line of inquiry concerns the *scope* of what RLVR can teach. Yue et al. (2025) find that standard RLVR raises pass@1 by amplifying rewarded traces already in the base model, while at large k the base model matches or exceeds the RL-trained model and coverage can shrink. Havrilla et al. (2024) find that RL fine-tuning fails to explore far beyond solutions already available after supervised fine-tuning. The model's *reasoning boundary*[1] therefore tends to plateau or even shrink after standard training. Yao et al. (2025) treat expansion and shrinkage as two stages of probability-mass reallocation: early exploitation can narrow coverage, and expansion requires training that continues into a later exploration stage.

What, then, about problems *operationally beyond* that boundary? In challenging domains, a non-trivial fraction of problems are *unsolvable*: every sampled rollout fails even under generous budgets (*e.g.*, pass@64 $= 0$). We find $\sim$7% of a 2,000-problem competition-math corpus is unsolvable for Qwen3-1.7B. These problems present a paradox: they mark the model's reasoning boundary and carry significant learning potential, yet under standard GRPO, uniform failure produces vanishing advantage and near-zero gradient.

*This work was conducted independently of Amazon and does not relate to the author's position there.

[1]Following Yue et al. (2025), we use "reasoning boundary" to denote the set of problems a model can solve given sufficient sampling, operationalized as pass@k at large k.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 19639–19658
October 24-29, 2026 ©2026 Association for Computational Linguistics

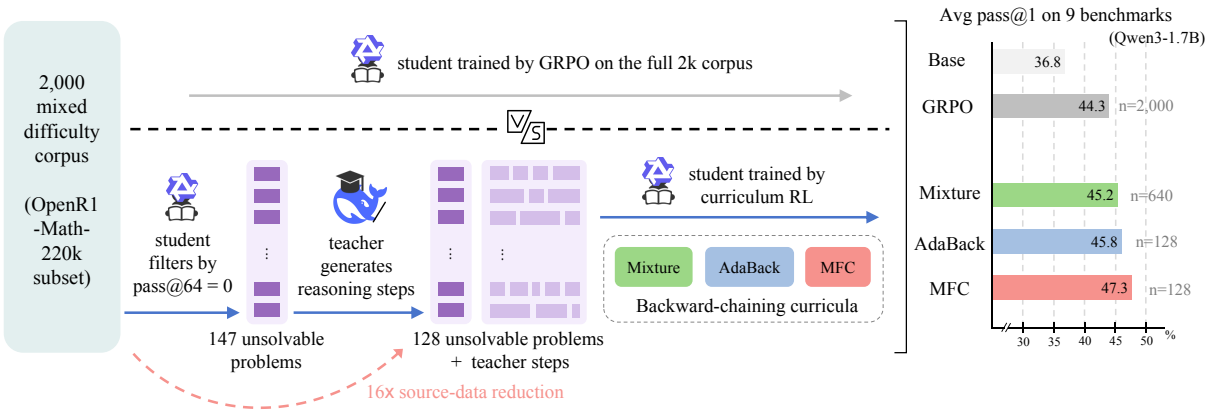


Figure 1: **Overview of our unsolvable-curriculum pipeline.** Top path: standard GRPO trained on the full 2,000-problem mixed-difficulty corpus. Bottom path: we filter the same corpus to its 147 unsolvable problems (pass@64 $= 0$ on the student), generate a step-structured trace with a stronger teacher, discard teacher failures (136 remain), and select the 128 shortest examples as the curriculum-training set. We compare three backward-chaining curricula: Mixture (R^3), AdaBack, and our **MFC**. MFC is designed for the unsolvable-only regime. With $\sim$16$\times$ less source data, the curriculum methods match or exceed the GRPO baseline on the cross-benchmark average (right; Qwen3-1.7B, 9 benchmarks).

This motivates our central question:

> *Can unsolvable problems be **unlocked**[2] for RLVR, and can they alone drive data-efficient reasoning improvement and frontier expansion?*

We answer affirmatively. Our key insight is that a partial reasoning trace from a stronger teacher transforms a single unsolvable problem into a *family of states* with graded difficulty, indexed by a guidance level $\rho \in [0, 1]$ controlling how much of the teacher's reasoning is revealed. A backward-chaining curriculum over ρ can then be utilized to progressively withdraw guidance toward unaided solving.

Figure 1 summarizes the pipeline. Training on **128 unsolvable problems** with backward-chaining curricula (Amani et al., 2026) matches or exceeds GRPO on the full 2,000-problem corpus ($\sim$16$\times$ data reduction) on the nine-benchmark average for both base models, and *substantially expands* the reasoning boundary (AIME pass@k improves at every k up to 256).

We further observe that existing backward-chaining curricula are not optimized for the unsolvable-only regime. Every training step at guidance level $\rho > 0$ occurs under a distribution that differs from the test-time distribution ($\rho = 0$), and generic curricula have no explicit mechanism to maximize training at $\rho = 0$. We formalize this distribution-shift cost in §3.2 and propose **Monotone Frontier Curriculum (MFC)**, a method that maintains a per-sample monotone non-increasing guidance frontier and consistently outperforms existing curricula.

Our contributions are as follows:

- We demonstrate that **teacher-guided curriculum learning unlocks unsolvable problems for RLVR**. Training on 128 such problems achieves data efficiency comparable to or exceeding GRPO on 2,000 mixed-difficulty problems.
- We show that this paradigm **expands the reasoning boundary** (pass@k gains at every k up to 256), contrasting with prior observations that RLVR narrows coverage (Yue et al., 2025).
- We propose **Monotone Frontier Curriculum (MFC)**, designed for the unsolvable-only regime, which maximizes unguided training through a monotone per-sample frontier and consistently outperforms existing curricula.

2 The Unsolvable Regime

We first establish the empirical foundation for our work: we formalize what it means for a problem to be *unsolvable* under RLVR (§2.1), verify that most of such problems remain inert throughout training

[2]Throughout this paper, *unlocking* refers to converting a problem that produces vanishing gradient under RLVR (due to uniform rollout failure) into one that yields non-trivial learning signal and meaningful group-relative advantage.

(§2.2), and show that teacher guidance can unlock them as a rich source of learning signal (§2.3).

2.1 Unsolvable Problems and Vanishing Gradient Signal

GRPO (Shao et al., 2024) trains a policy π_θ by sampling n rollouts per prompt and computing a group-relative advantage $A_i = (r_i - \mu_g)/(\sigma_g + \epsilon)$. When every rollout returns the same reward, $\sigma_g = 0$ and all advantages vanish, producing no gradient. For binary rewards this occurs whenever a group is uniformly correct or uniformly incorrect.

Definition 1 (Unsolvable problem)**.** *For a policy π_0 and a sampling budget N, a problem s is N-*unsolvable *if* $\text{pass@}N(\pi_0, s) = 0$*, i.e., no correct solution is found among N independent rollouts.*

The budget N should be large enough relative to the GRPO per-prompt rollout count n so that N-unsolvability reliably predicts uniform failure within each training group. We use $n = 8$ rollouts per group and set $N = 64$: a problem with no success in 64 attempts is overwhelmingly likely to produce $n = 8$ failures at almost every training step. From our 2,000-problem competition-mathematics corpus (details in Appendix C), approximately 7% of problems (147 total) are 64-unsolvable for Qwen3-1.7B. These problems potentially mark the model's reasoning boundary, yet they contribute negligible gradient signal under GRPO.

2.2 The Waste Hypothesis

One might expect training to gradually render some unsolvable problems productive, so that they begin generating learning signals. We test this by tracking the 147 unsolvable problems across all checkpoints of a 1,200-step GRPO baseline on Qwen3-1.7B (the same run described in §4), evaluating pass@8 at each checkpoint.

As shown in Figure 2, pass@8 rises to ∼24% by step 300–400 then plateaus; at step 1,200, more than 75% of initially unsolvable problems still yield zero successes. The small fraction that becomes learnable can be primarily explained by the model's modest capability growth during training. This confirms the **waste hypothesis**: under GRPO, the vast majority of unsolvable problems are *structurally excluded* from learning throughout training.

2.3 Unlocking via Teacher Guidance

If the bottleneck is uniform failure, a natural remedy is to reduce effective difficulty so that some succeed. We employ DeepSeek-V3.2 (DeepSeek-AI et al., 2025b,a) as the teacher model throughout this work; for each unsolvable problem s, we prompt the teacher model to generate a structured reasoning trace $T(s)$ consisting of discrete steps (details in Appendix G). A *guidance level* $\rho \in [0, 1]$, then, controls the fraction of this trace revealed to the student. Letting $n_{\text{steps}}(s)$ denote the number of steps in $T(s)$, we expose the first g steps:

$$\begin{aligned} g &= \text{round}(\rho \cdot n_{\text{steps}}(s)), \\ s_\rho &= s \cup \{\text{first } g \text{ steps of } T(s)\}. \end{aligned} \tag{1}$$

The student receives the augmented prompt s_ρ in which the teacher's partial trace is presented as auxiliary context (a "hint") and then generates a complete solution from scratch (see Appendix G for templates).

Table 1 shows the *unlock curve*: pass@8 as a function of ρ on base Qwen3-1.7B over the 147 unsolvable problems. We measure pass@8 because it mirrors the GRPO rollout regime ($n = 8$): a problem with pass@8 > 0 yields non-trivial group-relative advantage.

The curve rises sharply. pass@8 increases from 2.2% at $\rho = 0$ to 34.0% at $\rho = 0.25$ and 85.4% at $\rho = 1$, with a broad learnable band in $\rho \in [0.25, 1]$. The 2.2% residual means that the per-problem success probability is small but not strictly zero, which remains consistent with the finite-sample

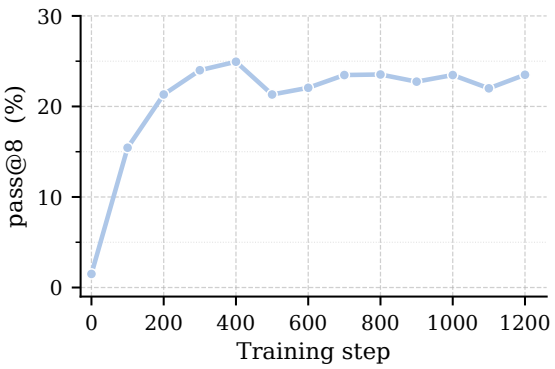


Figure 2: Average pass@8 on the 147 initially unsolvable problems (pass@64 = 0) across GRPO training checkpoints.

Guidance ρ	0	0.25	0.50	0.75	1.00
pass@8	2.2%	34.0%	53.6%	72.3%	85.4%

Table 1: pass@8 of Qwen3-1.7B on the 147 unsolvable problems at varying guidance levels ρ (averaged over 4 independent evaluations).

pass@64 = 0 filter. Teacher guidance converts the otherwise inert population into one that produces non-trivial advantage at every step, the operational meaning of *unlocking*.

Each unsolvable problem thus becomes a *family of training states* indexed by ρ, naturally suggesting a *backward-chaining curriculum*: begin at high ρ where the student reliably succeeds, and progressively withdraw guidance toward $\rho = 0$, where the student must solve the problem entirely on its own. We formalize this paradigm in the next section.

3 Curriculum Methods for Unsolvable Problems

3.1 The Backward-Chaining Family

A curriculum is *backward-chaining* if it places non-trivial training mass at $\rho > 0$ and progressively shifts that mass toward $\rho = 0$. Three canonical sub-families occupy this design space.

Staged curricula. A staged curriculum maintains a single global schedule $\rho_{\text{global}}(t)$ advanced when an aggregate success threshold is met (Salimans and Chen, 2018). In practice, curriculum performance is dominated by scheduler design, with no settled best practice for when and how to advance the global ρ. Vanilla staged reverse curricula can exhibit sharp drops at stage transitions and degradation of earlier-stage skill (Xi et al., 2024). For the experiments of §4 we therefore do not include a staged baseline.

Mixture curricula. A mixture curriculum bypasses the scheduler by expanding the per-sample state space: each problem is replicated at M discretized ρ-levels, and a single training pass uniformly samples across all levels. For the experiments of §4 we use R^3 (Xi et al., 2024) as the Mixture baseline with $M = 5$ levels ($\rho \in \{0, 0.25, 0.5, 0.75, 1\}$), expanding the 128 unsolvable problems into a 640-row dataset trained with standard GRPO.

Per-sample adaptive curricula. A per-sample adaptive curriculum maintains one schedule per sample and updates it based on rollout outcomes. The reference instantiation is AdaBack (Amani et al., 2026), a per-sample adaptive method that we compare with the R^3-style mixture below. Originally proposed over a continuous guidance level, AdaBack is adapted here to share MFC's (§3.3) discrete step lattice,[3] ensuring both methods sample from the same space (Appendix B). Per sample s, AdaBack maintains an integer interval $[g_{\min}(s), g_{\max}(s)] \subseteq \{0, \dots, n_{\text{steps}}(s)\}$ initialized to the full range. The first query uses $\rho = 0.5$, and subsequent visits sample $g_{\text{used}} \sim \text{Uniform}\{g_{\min}, \dots, g_{\max}\}$. AdaBack then updates the interval *bidirectionally*: a success ($\bar{r} \geq \tau$) sets $g_{\max} \leftarrow g_{\text{used}}$ and resets $g_{\min} \leftarrow 0$; a failure ($\bar{r} < \tau$) raises $g_{\min} \leftarrow g_{\text{used}}$. This implements a binary search for the per-sample *learnable band* around which AdaBack stabilizes.

3.2 The Hidden Cost in the Unsolvable-Only Regime

The unsolvable-only regime exposes a hidden inefficiency shared by all three methods above. We make this precise by separating the curriculum's training-time objective from the test-time objective it ultimately serves.

The training objective at step t under curriculum $\mathcal{C}$ samples $(s, \rho) \sim \mathcal{C}^{(t)}$ and conditions the policy on the augmented prompt s_ρ of Eq. (1):

$$J_{\mathcal{C}}^{(t)}(\theta) \;=\; \mathbb{E}_{(s,\rho)\sim\mathcal{C}^{(t)}}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|s_\rho)}\big[r(s,y)\big], \quad (2)$$

where $r(s, y)$ is based on the *original* problem s but the policy sees s_ρ whenever $\rho > 0$. The test-time objective is the unguided case

$$J_0(\theta) \;=\; \mathbb{E}_{s\sim\mathcal{D}}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|s)}\big[r(s,y)\big]. \quad (3)$$

Whenever $g(s, \rho) > 0$, the curriculum induces a *distribution shift* between training and test prompts. Improving $J_{\mathcal{C}}^{(t)}$ does not automatically improve J_0: augmented visits transfer to the unguided regime through parameter sharing, but the transfer is sign-indeterminate in general (Appendix F). A curriculum that aims to improve J_0 should therefore allocate training mass to the unguided regime.

We capture this via the **unguided training mass**

$$M_0^{(t)} \;:=\; \Pr_{(s,\rho)\sim\mathcal{C}^{(t)}}\big[g(s,\rho) = 0\big], \quad (4)$$

the in-batch fraction of visits at guidance length zero. Higher $M_0^{(t)}$ means more compute on prompts matching the test distribution. Appendix F provides the formal first-order decomposition of J_0 motivating this choice.

[3] From here on we operate on the integer $g \in \{0, \dots, n_{\text{steps}}(s)\}$ of Eq. (1); the continuous ρ is retained only for narrative description.

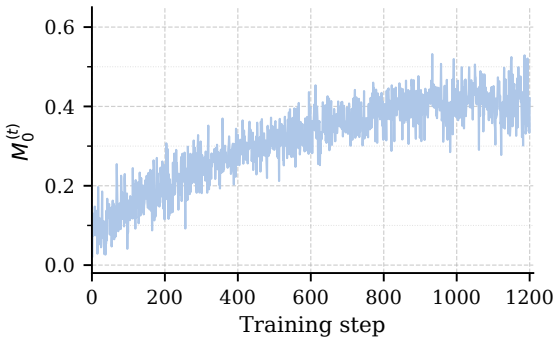


Figure 3: Unguided training mass $M_0^{(t)}$ of AdaBack on the 128-problem unsolvable corpus (Qwen3-1.7B, the same run in §4). $M_0^{(t)}$ rises initially but plateaus once per-sample intervals settle in the learnable band.

This surrogate matters more in the unsolvable-only regime than in mixed-difficulty training. In a mixed corpus, many samples are already feasible at $\rho = 0$, so $M_0^{(t)}$ rises naturally. In an unsolvable-only corpus, samples produce near-zero feasibility at $\rho = 0$ initially, and none of the existing curricula carry an explicit mechanism to drive descent toward $\rho = 0$ as the policy improves. AdaBack, for instance, binary-searches the per-sample *learnable band*, which for an unsolvable problem typically remains above $\rho = 0$ throughout training even as the policy improves substantially, so $M_0^{(t)}$ plateaus. Figure 3 confirms this empirically: AdaBack's $M_0^{(t)}$ rises to $\sim$0.40 by step 800 and remains essentially flat over the remaining steps.

3.3 Monotone Frontier Curriculum (MFC)

The analysis above identifies a need for a curriculum that can continue increasing its unguided training mass as the policy improves. MFC implements this goal by lowering a sample's guidance frontier only after the student succeeds with less guidance. As a result, each sample's unguided sampling probability increases monotonically with demonstrated competence, and sampled ρ shifts toward 0. It restructures AdaBack's update around an asymmetric principle: *success commits, failure is forgotten*. AdaBack raises $g_{\min}$ on failure. MFC keeps only the success-side ratchet, committing $g_{\text{curr}} \leftarrow g_{\text{used}}$ on a successful visit if smaller, while failures leave the per-sample state untouched and still contribute to the GRPO advantage. With $g_{\min}$ removed, the state collapses to a single integer $g_{\text{curr}}(s)$ that is monotonically non-increasing in t.

This asymmetry suits the unsolvable-only regime: with a rapidly evolving policy, failure at low ρ today is weak evidence for failure tomorrow, whereas success is firm evidence that a lower frontier is feasible. The monotone descent propagates directly to the surrogate: each successful visit at a lower g_{used} permanently raises MFC's per-sample on-target probability $1/(g_{\text{curr}}(s)+1)$ via the discrete-uniform draw of Algorithm 1. Once $g_{\text{curr}}(s) = 0$, all visits to that sample are at $\rho = 0$ (equivalently, $\rho_{\max}(s) := g_{\text{curr}}(s)/n_{\text{steps}}(s) = 0$).

Algorithm 1 gives the full procedure with a single hyperparameter τ (success threshold); we use $\tau = 0.5$ throughout.

Algorithm 1 Monotone Frontier Curriculum (MFC)

Require: Training corpus $\mathcal{D}$; for each prompt $s \in \mathcal{D}$, a teacher trace $T(s)$ given as an ordered list of $n_{\text{steps}}(s)$ reasoning steps; success threshold τ; rollouts per prompt n.

Initialization: $g_{\text{curr}}(s) \leftarrow n_{\text{steps}}(s)$ for all $s \in \mathcal{D}$.

for each visit to $s \in \mathcal{D}$ during GRPO training **do**
  $g_{\text{used}} \sim \text{Uniform}\{0, 1, \ldots, g_{\text{curr}}(s)\}$
  Roll out n trajectories from s with the first g_{used} steps of $T(s)$ prepended as hint
  Compute mean reward $\bar{r}$ and apply the GRPO update
  if $\bar{r} \geq \tau$ **and** $g_{\text{used}} < g_{\text{curr}}(s)$ **then**
    $g_{\text{curr}}(s) \leftarrow g_{\text{used}}$
  end if
end for

4 Experiments

4.1 Setup

Base models. We use **Qwen3-1.7B** (Yang et al., 2025) as the primary base model and **Qwen3-0.6B** as a smaller-scale replicate, both from the released base (non-instruct) checkpoint. Both models belong to the Qwen model family (Yang et al., 2024, 2025) and use the same tokenizer.

Datasets. All training and in-domain evaluation data are subsets of OpenR1-Math-220k (Hugging Face, 2025), derived through the quality-filter process of Appendix C. We use a 2,000-problem mixed-difficulty training set (**OR1-2k**) and a disjoint 200-problem held-out test set (**OR1-200**). The unsolvable training pool is built directly from OR1-2k: we evaluate Qwen3-1.7B at pass@64 and

Base Model	Method	In-domain		Math OOD					Non-math OOD		Avg.
		OR1-200	Uns-22	MATH-500	AMC23	AIME24	AIME25	AIME26	SciBench	GPQA-D	
Qwen3-1.7B	BASE	47.5	0.6	83.5	62.8	23.2	22.6	22.1	34.1	34.9	36.8
	GRPO	**68.6**	23.9	**89.5**	73.8	26.7	26.3	21.3	34.5	33.8	44.3
	MIXTURE	66.9	25.6	88.2	72.2	29.8	25.8	27.1	36.3	35.0	45.2
	ADABACK	67.2	29.5	89.1	72.5	30.0	**29.2**	26.3	34.1	34.1	45.8
	MFC	68.2	**31.8**	89.1	**75.0**	**32.5**	28.8	**27.2**	**37.1**	**36.1**	**47.3**
Qwen3-0.6B	BASE	28.8	0.0	67.1	38.1	4.2	11.3	8.8	18.4	29.3	22.9
	GRPO	**51.1**	16.5	**78.8**	50.0	11.7	15.4	8.8	21.2	28.3	31.3
	MIXTURE	45.4	19.3	74.4	50.0	9.6	14.6	9.6	21.5	30.6	30.6
	ADABACK	49.0	22.7	75.5	51.3	10.8	18.8	12.5	**23.0**	31.7	32.8
	MFC	50.4	**23.3**	76.8	**52.2**	**13.3**	**19.6**	**14.2**	22.8	**33.1**	**34.0**

Table 2: Main result. pass@1 (%, average of 8 completions at temperature 1) on nine benchmarks for two base models. **Bold** marks the column-wise best and underline the column-wise second-best, both computed *within* each base-model block.

retain the 147 problems with pass@64 $= 0$, generate teacher traces with DeepSeek-V3.2 and discard problems on which the teacher itself fails (136 remaining), and select the 128 shortest by combined question-plus-trace length to bound prompt budget, yielding **Uns-128**. The same Uns-128 is reused unchanged when training Qwen3-0.6B (a fortiori unsolvable for the weaker sibling). The analogous extraction on OR1-200 yields **Uns-22**, an in-domain unsolvable evaluation subset. We release the full OR1-2k, OR1-200, Uns-128, and Uns-22 records, together with the Uns-128 teacher traces, in a public Hugging Face dataset repository.[4]

Algorithms. For each base model we compare a *Base* (no-RL) reference and four trained algorithms, all run for 1,200 GRPO steps under matched compute: **(1) GRPO** on the full OR1-2k corpus with no teacher guidance; **(2) Mixture Curriculum** (R^3-style; §3.1) on Uns-128 expanded into a 640-row dataset at $M = 5$ uniform ρ-levels; **(3) AdaBack** on Uns-128 with the discrete adaptation of §3.1 ($\tau = 0.5$, default initial $\rho = 0.5$); and **(4) MFC** (ours) on Uns-128 with $\tau = 0.5$ (Algorithm 1). All teacher-guided runs provide the partial trace as a hint within the prompt (§2.3); all four share an identical GRPO core (batch size 128, $n = 8$ rollouts at temperature 1, 1e−6 learning rate, low-variance KL with coefficient 1e−3, 8,192-token response cap) augmented with two DAPO (Yu et al., 2025) stabilizers: *clip-higher* and an *overlong-response penalty*. Full hyperparameter tables and launcher pointers are in Appendix B.

[4] https://huggingface.co/datasets/yukangzhu/unlocking-the-unsolvable

Benchmarks. Evaluation spans **nine** benchmarks in three groups: two *in-domain math* sets (**OR1-200** and **Uns-22**, the latter directly probing in-domain reasoning-boundary lift); five *out-of-distribution math* sets (MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC23[5], AIME24/25/26[6]); and two *out-of-distribution non-math reasoning* sets (SciBench (Wang et al., 2024b) and GPQA-Diamond (Rein et al., 2024)). All scores are pass@1 averaged over 8 completions at temperature 1; correctness is judged by Math-Verify[7], a rule-based verifier that normalizes and compares model outputs against ground-truth answers.

4.2 Main Results

Table 2 reports pass@1 on all nine benchmarks for both base models and the four trained algorithms, with the un-trained *Base* reference for context. We summarize three results from Table 2:

1. 128 **unsolvable problems match or exceed** 2,000 **mixed.** On Qwen3-1.7B, AdaBack and MFC stay within 1.4 pp of the GRPO 2k baseline on OR1-200 and MATH-500. On Qwen3-0.6B, the corresponding gaps range from 0.7 to 3.3 pp. Both methods exceed GRPO on Uns-22 and on the average across AIME24/25/26 (e.g. +7.9 / +6.8 pp for MFC on Uns-22), with a ∼16× cut in source data. Mixture is competitive on Qwen3-1.7B but trails GRPO on the Qwen3-0.6B average

[5] https://huggingface.co/datasets/zwhe99/amc23

[6] https://huggingface.co/datasets/hendrydong/aime24, https://huggingface.co/datasets/math-ai/aime25, https://huggingface.co/datasets/math-ai/aime26

[7] https://github.com/huggingface/Math-Verify

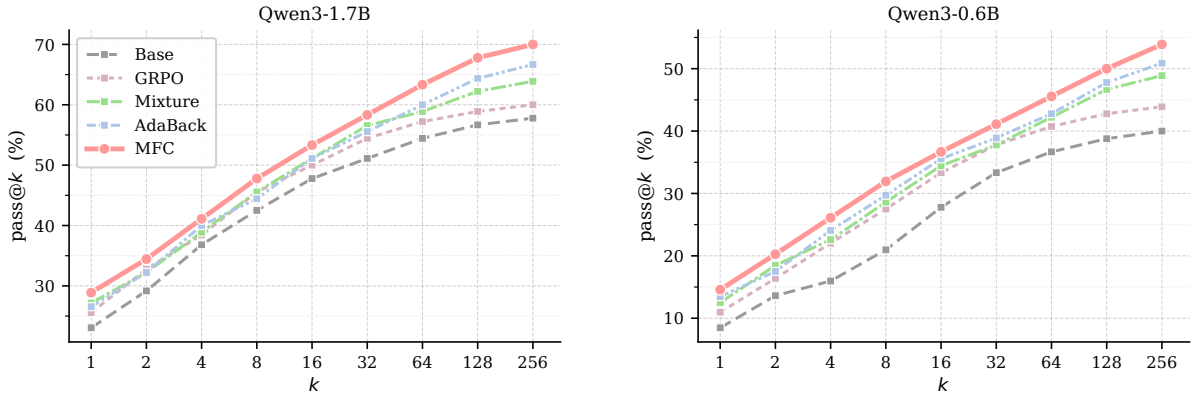


Figure 4: Reasoning boundary measured by pass@k on the union of AIME24/25/26 (90 problems), $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ on a $\log_2$ axis. Left: Qwen3-1.7B; right: Qwen3-0.6B. The unsolvable-curriculum methods (Mixture, AdaBack, MFC) widen the gap over both *Base* and GRPO at large k, with MFC uniformly best.

(30.6 vs 31.3), so uniform-ρ mixing alone is not enough at the smaller scale.

2. **MFC has the best cross-benchmark average on both base models.** Its averages are 47.3 and 34.0. On the AIME24/25/26 average it improves over GRPO by +4.7 / +3.7 pp on Qwen3-1.7B / 0.6B, and over AdaBack by +1.0 / +1.7 pp.

3. **The ordering MFC > AdaBack > GRPO holds on both base models** on the cross-benchmark average, on Uns-22, and on the AIME average. Mixture and GRPO trade positions across base models. MFC's lead over AdaBack holds on both models, so the advantage of monotone frontier descent (§3.3) is not specific to one model size.

The non-math OOD benchmarks (SciBench, GPQA-Diamond) tell a complementary story: training on 128 math-only unsolvable problems does not degrade non-math reasoning and shows mild positive transfer for MFC (+1.2 to +4.4 pp over Base across the two base models), suggesting that the model's improved reasoning capability potentially generalizes beyond mathematics.

4.3 Reasoning Boundary Expansion

We evaluate whether unsolvable-curriculum training expands the model's reasoning *boundary*. We measure pass@k on the union of AIME24/25/26 (90 problems total) for $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$, using the same five rows of Table 2 (*Base*, GRPO, Mixture, AdaBack, MFC) on each base model.

Plain GRPO shifts the frontier modestly. On Qwen3-1.7B, pass@64 rises from 54.4% (Base) to 57.2% (+2.8 pp), and pass@256 rises from 57.8% to 60.0% (+2.2 pp). On Qwen3-0.6B, the corresponding gains are +4.1 pp at pass@64 and +3.9 pp at pass@256. These gains still trail every teacher-guided method, consistent with prior reports (Yue et al., 2025) that standard RL primarily re-weights existing solution paths. Mixture, AdaBack, and MFC each add several further points at large k, and their gains over Base tend to be larger at high k. Large k is the setting that most directly measures reasoning coverage. MFC is highest on both base models at every k. At pass@256, it reaches 70.0% on Qwen3-1.7B and 53.9% on Qwen3-0.6B, exceeding GRPO by +10.0 pp for both models. Its lead over AdaBack widens at large k, reaching +3.0 pp at pass@256 for Qwen3-0.6B and +3.3 pp for Qwen3-1.7B. Together with the analysis in §3.2, the higher unguided training mass accumulated by MFC is associated with broader reasoning coverage at large k.

4.4 Data Efficiency

The four algorithms span a ∼16× range in source-data size: 2,000 problems for GRPO, 640 rows (128 × 5) for Mixture, and 128 problems for AdaBack and MFC, all under matched compute and matched 1,200 GRPO steps. As summarized in the right panel of Figure 1 and Table 2, AdaBack and MFC match or exceed the GRPO 2k cross-benchmark average on both base models. Mixture exceeds GRPO on Qwen3-1.7B but trails it on Qwen3-0.6B. Under matched compute, MFC on Uns-128 attains a higher cross-benchmark average

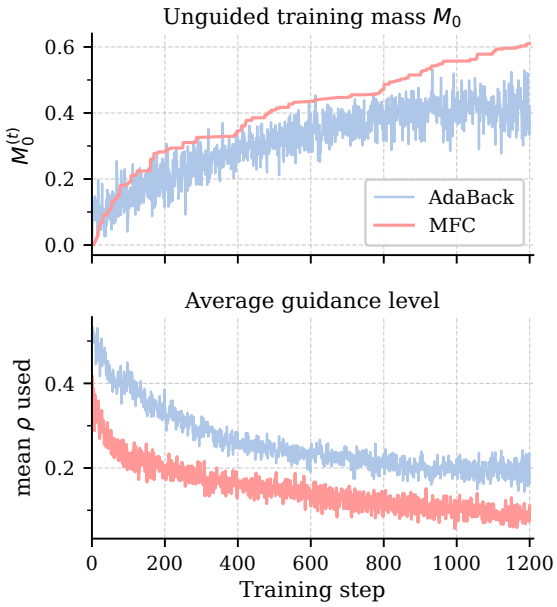


Figure 5: Curriculum dynamics of MFC vs. AdaBack on Uns-128 (Qwen3-1.7B). Top: unguided training mass $M_0^{(t)}$; bottom: average guidance level used per step. MFC continues to drive training toward the unguided regime throughout the 1,200-step run, whereas AdaBack largely plateaus after step ∼800.

than GRPO on OR1-2k while using ∼16× fewer source problems.

5 Analysis

5.1 Curriculum Dynamics

We now verify that this monotone update raises the unguided training mass $M_0^{(t)}$ (Eq. (4)) and lowers the sampled guidance level. Figure 5 compares MFC and AdaBack on Uns-128 (Qwen3-1.7B) across $M_0^{(t)}$ and mean_rho_used, the batch-averaged guidance level ρ actually sampled at each step (lower is closer to the test-time regime). By step 1,200, MFC reaches a smoothed $M_0^{(t)} \approx 0.60$ versus AdaBack's ≈ 0.40, and drives mean_rho_used down to ≈ 0.09 versus AdaBack's ≈ 0.20. The pattern is consistent: MFC's monotone frontier ratchets $\rho_{\max}$ downward on every successful visit, progressively shifting the sampling distribution toward $g = 0$; AdaBack's bidirectional update stabilizes once a learnable band is established and has no explicit mechanism to push further. The same separation holds on Qwen3-0.6B (Appendix D.2).

Appendix F decomposes ΔJ_0 into an unguided term weighted by $M_0^{(t)}$ and an augmented term whose sign is unconstrained. MFC's higher M_0 corresponds to a larger share of compute spent on prompts with the test-time input format. Empirically, this increase is associated with a higher cross-benchmark average and broader reasoning coverage (Figure 4).

Method	Avg. (9 benchmarks)
BASE	36.8
SFT (FULL TRACE)	39.8
ADABACK	45.8
MFC	**47.3**

Table 3: Cross-benchmark average pass@1 (%) on Qwen3-1.7B. SFT uses the same Uns-128 corpus and full teacher traces; curriculum RL methods use the same data under adaptive guidance. Per-benchmark breakdown in Appendix E.

5.2 SFT Baseline vs. Curriculum RL

A natural concern is that curriculum RL on teacher traces is effectively expensive distillation. We test this with a supervised-fine-tuning (SFT) baseline: Qwen3-1.7B trained on the same Uns-128 problems via next-token prediction on the full teacher trace ($\rho = 1$), using the identical $\rho = 0$ prompt template as the RL evaluation prompt. We train for 50 epochs (lr 1e−5, cosine schedule) and select the step-40 checkpoint where loss has converged; full configuration in Appendix E.

Table 3 shows that SFT improves over Base by only +3.0 pp on average, far short of AdaBack (+9.0) and MFC (+10.5). SFT memorizes a single teacher path per problem and converges quickly (under 50 epochs, see Appendix E) on this small corpus; curriculum RL, by contrast, drives ρ to 0 under reward feedback and forces the policy to discover its own solutions. The curriculum-RL gains are therefore not attributable to teacher-trace distillation: the same 128 problems and the same teacher information yield qualitatively different outcomes under SFT versus curriculum RL.

6 Related Work

Problems beyond the RLVR frontier. Studies of reasoning boundaries find that standard RLVR often amplifies solution paths already available to the model, while many problems remain outside its sampled coverage (Yue et al., 2025; Havrilla et al., 2024). Uniform failure creates an optimization problem because a GRPO group with identical rewards has no group-relative advantage. DAPO dynamic sampling removes groups with accuracy

zero or one when constructing a training batch (Yu et al., 2025). RL-ZVP keeps zero-variance groups and supplies an entropy-guided token-level advantage (Le et al., 2026). SwS identifies weaknesses from persistent training failures and synthesizes new questions around the associated concepts (Liang et al., 2025). Adaptive Difficulty Curriculum Learning (ADCL) periodically re-estimates problem difficulty and reorders upcoming batches as model capability changes (Zhang et al., 2025). For an unsolvable-only pool, reordering alone does not change uniformly failed rollout groups. Our setting keeps the original problems selected by $\text{pass@64} = 0$. Partial teacher traces turn each problem into guided states with mixed rollout outcomes, allowing the curriculum to train on the same fixed corpus.

Backward-chaining over partial solutions. When a solution trace is available, the amount of revealed reasoning becomes a curriculum variable. R^3 moves the generation starting point through a correct demonstration and mixes examples from several stages (Xi et al., 2024). In our setup the trace is auxiliary context, and the student generates a complete solution from scratch. AdaBack adapts the revealed portion per sample through bidirectional updates to a feasible interval (Amani et al., 2026). SEELE adjusts hint length to keep rollout accuracy near a target level using item response theory (Li et al., 2025). MFC keeps a single upper frontier and lowers it only after success. This update is designed to increase unguided training mass when every original prompt is initially infeasible.

Teacher traces and data-efficient RLVR. Data-centric studies show that hard problems, high reward variance, and even carefully chosen single examples can be especially useful for RLVR (Pikus et al., 2025; Foster et al., 2025; Wang et al., 2025). Our study considers the endpoint where the selected problems have no sampled success before guidance. Teacher-generated solutions are also widely used as supervised targets or to bootstrap reasoning training (Hinton et al., 2015; Zelikman et al., 2022; Gulcehre et al., 2023; Zhang et al., 2024). Here the teacher trace enters the prompt as context for an RL objective, and the student still produces a complete solution. The SFT comparison in §5.2 isolates this use of the same Uns-128 traces as supervised targets. Appendix A reviews the broader data-selection, curriculum-learning, and teacher-guidance literature.

7 Conclusion

We showed that *unsolvable problems*, defined by $\text{pass@64} = 0$ for the base model, can be turned into productive RLVR training signal via a stronger teacher's structured trace combined with a backward-chaining curriculum. Training on only 128 such problems matches or exceeds a 2,000-problem GRPO baseline on the nine-benchmark average for both base models ($\sim$16$\times$ data efficiency) and expands the reasoning boundary ($\text{pass@}k$ gains up to $k = 256$ on AIME24/25/26).

Within this regime we further identified a distribution-shift cost inherent to guided curricula and proposed **Monotone Frontier Curriculum (MFC)**, whose per-sample frontier descends monotonically under a success-only update. MFC outperforms both the GRPO baseline and existing backward-chaining curricula on the cross-benchmark average for both base models. Our findings suggest that teacher-guided curricula, paired with the right scheduling mechanism, can unlock substantial reasoning gains from a remarkably small pool of hard problems.

Limitations

Our work relies on a single fixed teacher model (DeepSeek-V3.2) to generate all reasoning traces, and the quality of the resulting curriculum is inevitably tied to the quality of these traces. We do not study how teacher capability interacts with curriculum design. A stronger teacher may raise the attainable $M_0^{(t)}$ per sample, and it may also introduce reasoning patterns whose hints transfer less cleanly to a smaller student.

The framework is trained only on mathematical reasoning with single-turn LLM outputs. SciBench and GPQA-Diamond provide out-of-domain probes, but they do not constitute a study of scientific reasoning. Transfer to code generation (Li et al., 2022) or formal theorem proving is untested. These domains introduce distinct challenges: code requires multi-file coherence and execution-based verification, while scientific reasoning often lacks the discrete step structure our hint mechanism relies on. We have not applied the curriculum to *agentic* tasks (Jimenez et al., 2024; Zhou et al., 2024), which need sequential environment actions and multi-step tool-use traces.

References

Mohammad Hossein Amani, Aryo Lotfi, Nicolas Baldwin, Samy Bengio, Mehrdad Farajtabar, Emmanuel Abbe, and Robert West. 2026. RL for reasoning by adaptively revealing rationales. In *The Fourteenth International Conference on Learning Representations*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. Process reinforcement through implicit rewards. *Preprint*, arXiv:2502.01456.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025a. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025b. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Nicolaus Foerster. 2025. LILO: Learning to reason at the frontier of learnability. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling. *Preprint*, arXiv:2308.08998.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Dac Lai, and Eunho Yang. 2026. No prompt left behind: Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *The Fourteenth International Conference on Learning Representations*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, and 7 others. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Ziheng Li, Zexu Sun, Jinman Zhao, Erxue Min, Yongcheng Zeng, Hui Wu, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Xu Chen, and Zhi-Hong Deng. 2025. Staying in the sweet spot: Responsive reasoning evolution via capability-adaptive hint scaffolding. *Preprint*, arXiv:2509.06923.

Xiao Liang, Zhong-Zhi Li, Yeyun Gong, Yang Wang, Hengyuan Zhang, Yelong Shen, Ying Nian Wu, and Weizhu Chen. 2025. Sws: Self-aware weakness-driven problem synthesis in reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *Preprint*, arXiv:2412.09413.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Benjamin Pikus, Pratyush Ranjan Tiwari, and Burton Ye. 2025. Hard examples are all you need: Maximizing grpo post-training under annotation budgets. *arXiv preprint arXiv:2508.14094*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Tim Salimans and Richard Chen. 2018. Learning montezuma's revenge from a single demonstration. *arXiv preprint arXiv:1812.03381*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. 2026. Towards high data efficiency in reinforcement learning with verifiable reward. In *The Fourteenth International Conference on Learning Representations*.

Stone Tao, Arth Shukla, Tse kai Chan, and Hao Su. 2024. Reverse forward curriculum learning for extreme sample and demo efficiency. In *The Twelfth International Conference on Learning Representations*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024b. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In *Forty-first International Conference on Machine Learning*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025. Reinforcement learning for reasoning in large language models with one training example. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le,

and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, peng sun, Tao Gui, and 2 others. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. In *Forty-first International Conference on Machine Learning*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Xinhao Yao, Lu Yu, Xiaolin Hu, Fengwei Teng, Qing Cui, Jun Zhou, and Yong Liu. 2025. The debate on rlvr reasoning capability boundary: Shrinkage, expansion, or both? a two-stage dynamic view. *arXiv preprint arXiv:2510.04028*.

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, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. 2025. Learning like humans: Advancing LLM reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6619–6633, Suzhou, China. Association for Computational Linguistics.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Preprint*, arXiv:2304.11277.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. 2026. 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)*, pages 46167–46192, San Diego, California, United States. Association for Computational Linguistics.

A Extended Related Work

Data-centric RLVR. The choice of training data profoundly impacts RLVR outcomes. Pikus et al. (2025) demonstrate that filtering to the hardest problems yields disproportionate gains over uniform sampling, while Foster et al. (2025) formalize the "learnability" criterion showing that maximizing reward variance is equivalent to optimizing expected improvement. Wang et al. (2025) push this further, showing a single well-chosen problem can unlock significant reasoning capability. Tang et al. (2026) study data efficiency for RLVR through offline subset selection based on diversity, influence, and moderate difficulty, together with online explorability-based rollout pruning, and Zhu et al. (2026) propose influence-guided data selection to accelerate training. Complementary work on reward signal design—process reward (Uesato et al., 2022; Wang et al., 2024a) versus outcome reward—also shapes effective data utilization. Our work identifies the extreme end of this spectrum—problems with *zero* feasibility—and shows they can be made productive through teacher guidance rather than filtering them away. The treatment of problems beyond the sampled frontier is discussed in §6.

Curriculum learning beyond partial traces. Curriculum learning orders or samples training experiences according to difficulty (Bengio et al., 2009). Reverse curricula in reinforcement learning start near successful states and move the initial state backward as the policy improves (Salimans and Chen, 2018; Tao et al., 2024). Other curricula act at the batch level by selecting or reordering problems, including schedules that track changes in model capability (Zhang et al., 2025). Our guidance lattice acts within each problem by controlling the portion of a trace supplied as context. Related studies of reasoning-boundary dynamics examine how prolonged RLVR interacts with exploration and coverage (Liu et al., 2025a; Yao et al., 2025).

Teacher traces as targets and as context. Knowledge distillation (Hinton et al., 2015) and imitation learning (Ross et al., 2011) train students against outputs or actions supplied by a teacher. Chain-of-thought prompting (Wei et al., 2022) shows that explicit reasoning steps can improve generation, while STaR (Zelikman et al., 2022) and process reward methods (Luo et al., 2024) use intermediate reasoning during training. Guo et al. (2025) demonstrate that supervised distillation from reasoning traces can produce capable models. Gulcehre et al. (2023) and Zhang et al. (2024) use generated solutions to bootstrap further training. STILL-2 distills long reasoning traces and then bootstraps from successful rollouts (Min et al., 2024). Our SFT baseline in §5.2 compares target-based training with the use of teacher traces as prompt context under curriculum RL.

B Implementation Details

This appendix specifies the training configuration shared by all RLVR runs in §4, the per-method curriculum hyperparameters, and the hardware setup. All training and rollout infrastructure is built on top of verl (Sheng et al., 2025); our curriculum algorithms (Mixture, AdaBack, MFC) are implemented as additional curriculum managers within the same verl-based stack, so that the four trained methods share an identical training and inference pipeline.

B.1 Shared GRPO Training Core

All four trained algorithms (GRPO, Mixture, AdaBack, MFC) on both base models share an identical GRPO training core (Table 4). Two modifications inherited from DAPO (Yu et al., 2025) are layered on top: an asymmetric PPO clip ratio ($\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$), and a length-based *overlong-response penalty* that linearly penalizes incorrect responses approaching the 8,192-token cap (buffer length 1024, penalty factor 1.0). The penalty extends the verifiable reward from $\{0, 1\}$ to $[-1, 1]$ but does not alter the group-relative advantage structure of GRPO. KL regularization is applied as an auxiliary loss term using a low-variance estimator (`kl_loss_type=low_var_kl`) rather than mixed into the reward.

Parameter	Value
Rollouts per prompt (n)	8
Training batch size	128
PPO mini-batch size	128 (full update)
Rollout temperature	1.0 (top-p 1.0)
Max prompt length	800 (GRPO 2k) / 2,500 (hint)
Max response length	8,192
Optimizer	AdamW, lr 1e−6 (constant)
KL loss coefficient	1e−3, low-var estimator
PPO clip ratio (low / high)	0.2 / 0.28
Entropy coefficient	0
Overlong-response penalty	enabled, buffer 1024, factor 1.0
Advantage normalisation	by group std (default)
Total GRPO steps	1,200
Validation interval	every 10 steps
Validation decoding	T=0.6, top-p=0.95, n=1

Table 4: Shared GRPO training core, identical across all four trained algorithms and both base models in §4.

B.2 Per-Method Curriculum Hyperparameters

Method-specific curriculum hyperparameters are listed in Table 5. Mixture has no learnable per-sample state; its only curriculum choice is the discretized ρ-level grid. AdaBack follows the public-recipe defaults; two parameters bear brief explanation. The *on-target valve* p_{zero} is the small fixed probability of overriding the sampled ρ with $\rho = 0$, introduced in the original AdaBack paper (Amani et al., 2026) to close the train-test distribution mismatch (we use $p_{\text{zero}} = 0.1$). The *minimum step delta* governs the smallest discrete change to the per-sample hint length per update. MFC reduces to a single hyperparameter, the success-commit threshold τ (Algorithm 1).

Discretization of AdaBack. We keep AdaBack's hyperparameters in the original continuous-ρ form for fidelity with the public recipe (Table 5), but the

Method	Parameter	Value
GRPO	training corpus	OR1-2k (2,000 problems)
	teacher guidance	none ($\rho \equiv 0$)
MIXTURE	training corpus	Uns-128 expansion (640 rows)
	ρ-level grid	$\{0, 0.25, 0.5, 0.75, 1\}$
ADABACK	training corpus	Uns-128
	success threshold τ	0.5
	first-query ρ	0.5
	on-target valve p_{zero}	0.1
	minimum step delta	1
MFC	training corpus	Uns-128
	success threshold τ	**0.5**
	initial $\rho_{\max}$	1.0

Table 5: Method-specific curriculum hyperparameters. The three teacher-guided methods all use hint-mode guidance with the discretization of §3.1.

actual rollout samples on the integer hint lattice so that AdaBack and MFC share an identical sampling space (§3.1). At each rollout, the continuous bounds $[\rho_{\min}(s), \rho_{\max}(s)]$ are mapped to integer bounds via $g = \text{round}(\rho \cdot n_{\text{steps}}(s))$, an integer $g_{\text{used}} \sim \text{Uniform}\{g_{\min}, \ldots, g_{\max}\}$ is drawn, and the bidirectional update is applied directly on the integer interval (the minimum step delta of 1 ensures each successful update strictly contracts it). This rules out the round-induced sampling bias at $g = 0$ that would otherwise systematically penalize AdaBack's measured $M_0^{(t)}$ relative to MFC's discrete-uniform draw.

B.3 Hardware and Wall-Clock

All RLVR runs use 8× NVIDIA RTX PRO 6000 (96 GB) GPUs; FSDP (Zhao et al., 2023) with parameter and optimizer sharding is used throughout (offload disabled). Rollouts are served by vLLM (Kwon et al., 2023) at 70% GPU memory utilization with prefix caching enabled. Dynamic batching uses `ppo_max_token_len_per_gpu` = 48,000 and `ppo_micro_batch_size_per_gpu` = 4. A single 1,200-step training run takes approximately 90 wall-clock hours for Qwen3-1.7B and approximately 70 hours for Qwen3-0.6B on the same 8× RTX PRO 6000 setup. Including all diagnostic and development runs, our total compute usage amounted to approximately **7,000** RTX PRO 6000 GPU hours.

C Training-Data Construction

This appendix details the construction pipeline for the four corpora used in §4.1: the mixed-difficulty training set **OR1-2k**, the held-out test set **OR1-200**, the 128-problem unsolvable training set **Uns-128**, and the held-out unsolvable subset **Uns-22**. All four are derived from the *train* split of OpenR1-Math-220k (Hugging Face, 2025) (93.7k problems) under a single, deterministic protocol with random seed 42.

Filtering and train/test split (→ OR1-2k, OR1-200). Starting from the 93.7k-problem train split, we apply a chain of quality filters to keep the corpus narrowly scoped to verifiable, single-answer competition-level mathematical reasoning problems (substantially harder than elementary benchmarks such as GSM8K (Cobbe et al., 2021)):

- drop problems where the released DeepSeek-R1 reference solution is judged incorrect by Math-Verify[8];
- drop multiple-choice problems (where the verifiable reward degenerates to chance);
- drop problems whose statement is too short or too long, and problems whose released reference solution is too long, to keep prompt and rollout budgets bounded;
- drop problems whose ground-truth answer string contains a comma (a strong indicator of multi-answer or list-valued problems that defeat exact-match scoring).

From the surviving pool we randomly draw 2,200 problems under seed 42 and split them into 2,000 training problems (**OR1-2k**) and 200 held-out test problems (**OR1-200**), keeping the splits disjoint. Difficulty composition is preserved by construction (random sampling), and the resulting train set spans the full pass@64 spectrum from trivial (pass@64 ≈ 1) to unsolvable (pass@64 = 0).

Unsolvable filtering (→ Uns-128). The unsolvable pool is derived only from OR1-2k:

1. Re-evaluate every problem in OR1-2k with Qwen3-1.7B at the response-length cap used at training time (8,192 tokens) and retain those with pass@64 = 0 (147 problems, ∼7% of the train set).

[8] https://github.com/huggingface/Math-Verify

2. Generate a structured reasoning trace for every retained problem with the DeepSeek-V3.2 (DeepSeek-AI et al., 2025b) teacher (the prompt template, structured-output schema, and step-extraction rule are described in Appendix G). Drop any problem on which the teacher itself fails the Math-Verify check, leaving 136 raw rows.

3. Sort the surviving problems by combined question-plus-trace character length and take the 128 shortest. This length-controlled selection bounds the maximum prompt length seen at training time and removes a small fraction of pathologically long traces; it is otherwise content-agnostic.

The resulting 128-problem set is **Uns-128**, the single unsolvable training corpus shared by Mixture, AdaBack, and MFC.

In-domain unsolvable test set (→ Uns-22). For evaluation purposes, we additionally extract the 22-problem subset of OR1-200 with $\mathrm{pass@64} = 0$ on Qwen3-1.7B, using the same 8,192-token response cap as the unsolvable filter above. This subset, **Uns-22**, is a strict held-out evaluation set: it shares no problems with Uns-128 and is never used for training or hyperparameter selection.

Reuse for Qwen3-0.6B. Uns-128 was filtered using Qwen3-1.7B's pass@64 but is reused unchanged when training Qwen3-0.6B. Because Qwen3-0.6B is a strictly weaker sibling from the same model family and training generation, every problem unsolvable for Qwen3-1.7B is at least as hard for Qwen3-0.6B. Sharing the unsolvable corpus across base models keeps the comparison apples-to-apples and avoids a confound where the smaller model would otherwise see an easier curriculum.

Data release. We release complete records for OR1-2k, OR1-200, Uns-128, and Uns-22 as a public Hugging Face dataset repository.[9] The four configs `or1_2k`, `uns128`, `or1_200`, and `uns22` can be loaded and used directly. Teacher traces appear only in the Uns-128 config. These files can be used for training or evaluation without mapping their identifiers back to OpenR1-Math-220k.

[9] https://huggingface.co/datasets/yukangzhu/unlocking-the-unsolvable

D Additional Experimental Details

D.1 Curriculum Metric Definitions

Throughout §5.1 we report two curriculum-level metrics, computed per training step:

- **Unguided training mass** $M_0^{(t)}$ (Eq. (4); logged as `frac_effective_zero` or `frac_at_zero`): the fraction of in-batch prompt visits whose discretized hint length $g(s, \rho) = 0$. Higher values indicate that a larger share of training compute is spent on prompts matching the test-time distribution.

- **Mean guidance level** mean_rho_used: the batch-averaged value of the guidance level ρ actually sampled at each step, i.e. $\frac{1}{|\mathcal{B}|} \sum_{(s,\rho)\in\mathcal{B}} \rho$. Lower values indicate the curriculum is closer to the unguided regime on average.

D.2 Qwen3-0.6B Curriculum Dynamics

Figure 6 replicates the curriculum-dynamics comparison of Figure 5 (main text, Qwen3-1.7B) on Qwen3-0.6B. The qualitative pattern is identical: MFC's $M_0^{(t)}$ continues to climb throughout training while AdaBack plateaus, and MFC's mean_rho_used descends well below AdaBack's. This confirms that MFC's structural advantage is not specific to the 1.7B scale.

D.3 Training and Validation Reward Dynamics

Figures 7 and 8 show the training reward and validation reward across all four methods on both base models.

Training reward (Figure 7). An important caveat: training reward is *not* directly comparable across methods. The curriculum mechanisms (Mixture, AdaBack, MFC) present each sample at a guidance level selected by their respective scheduling rules; higher ρ yields easier effective prompts and thus higher expected reward. GRPO, by contrast, trains on mixed-difficulty data without guidance. Consequently, the absolute level and the trajectory of training reward reflect the effective difficulty of the curriculum at each step, not the model's test-time capability.

Validation reward (Figure 8). Validation is always performed on OR1-200 without any teacher guidance ($\rho = 0$), making it a fair apples-to-apples

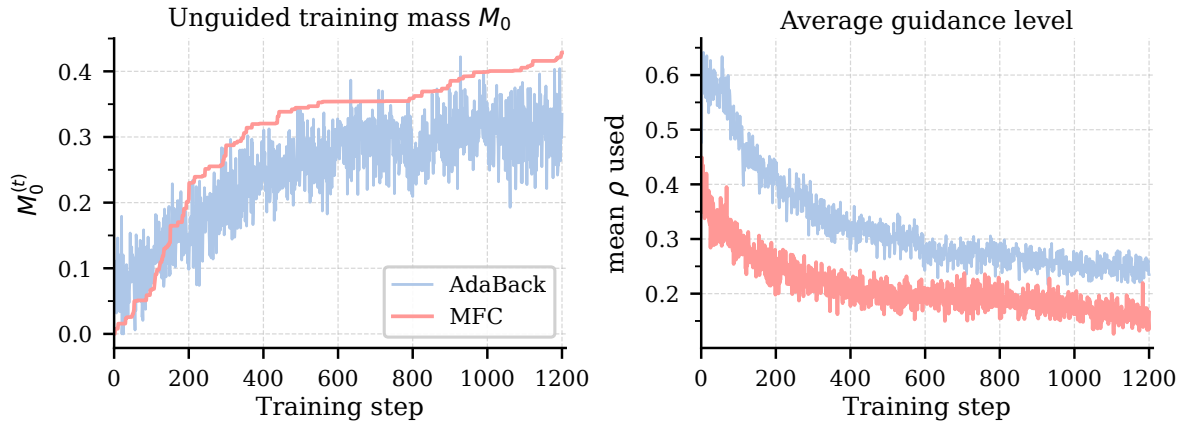


Figure 6: Curriculum dynamics of MFC vs. AdaBack on Uns-128 (Qwen3-0.6B). Left: unguided training mass $M_0^{(t)}$; right: average guidance level. The qualitative separation is identical to the 1.7B results (Figure 5).

comparison of unguided capability at each checkpoint. All four methods converge to broadly similar validation reward by step 1,200, but the curriculum methods reach this level faster (fewer wasted early steps) and MFC shows the steepest early rise, consistent with its higher $M_0^{(t)}$ allocating more on-target gradient compute from the start.

E SFT Baseline Details

This appendix provides full details for the SFT baseline of §5.2.

Data construction. Each of the 128 Uns-128 samples is formatted as a single-turn model conversation: the system and user messages are identical to the $\rho = 0$ prompt template used during curriculum RL evaluation, with no hints or guidance of any kind; the assistant message contains the full teacher reasoning wrapped in `<think>...</think>` tags followed by `\boxed{teacher_answer}`, exactly matching the generation format trained by curriculum RL. The `teacher_answer` field is recovered from the original training-pool metadata to ensure complete fidelity to the teacher model's own output.

Training configuration. We train Qwen3-1.7B (base) via full-parameter SFT (no LoRA) using FSDP on 2× NVIDIA RTX PRO 6000 GPUs. Key hyperparameters: global batch size 128 (entire dataset in one step), learning rate 1e−5 with cosine decay and 10% warmup, weight decay 0.01, gradient clipping at 1.0, bf16 mixed precision, max sequence length 2,048 tokens, 50 epochs (= 50 gradient steps). The 2,048-token cap is shorter than the 8,192-token response cap used by curriculum RL, but exceeds the full prompt-plus-trace length of every problem in Uns-128, so no truncation occurs in the SFT data. Total wall-clock time is approximately 15 minutes.

Checkpoint selection. Figure 9 shows the training loss curve. Since the validation set equals the training set for this 128-sample corpus, we show only training NLL. Loss levels off by step ∼35–40. We select step 40 as the evaluation checkpoint.

Per-benchmark results. Table 6 reports the full per-benchmark pass@1 for the SFT baseline (Qwen3-1.7B, step 40).

F Formal View of the Unguided Training Mass Surrogate

This appendix formalizes the on-target / augmented gradient decomposition that motivates the unguided training mass $M_0^{(t)}$ of Eq. (4) as the surrogate of choice in §3.2, and gives a careful comparison of MFC and AdaBack on $M_0^{(t)}$.

First-order Taylor decomposition. Let θ denote the current parameters and consider one gradient step against the curriculum objective $J_C^{(t)}$ of Eq. (2), $\theta' = \theta + \eta \nabla J_C^{(t)}(\theta)$ for a small step size $\eta > 0$. The first-order Taylor expansion of the test-time objective J_0 of Eq. (3) gives

$$J_0(\theta') - J_0(\theta) = \eta \left\langle \nabla J_0(\theta), \nabla J_C^{(t)}(\theta) \right\rangle + O(\eta^2). \quad (5)$$

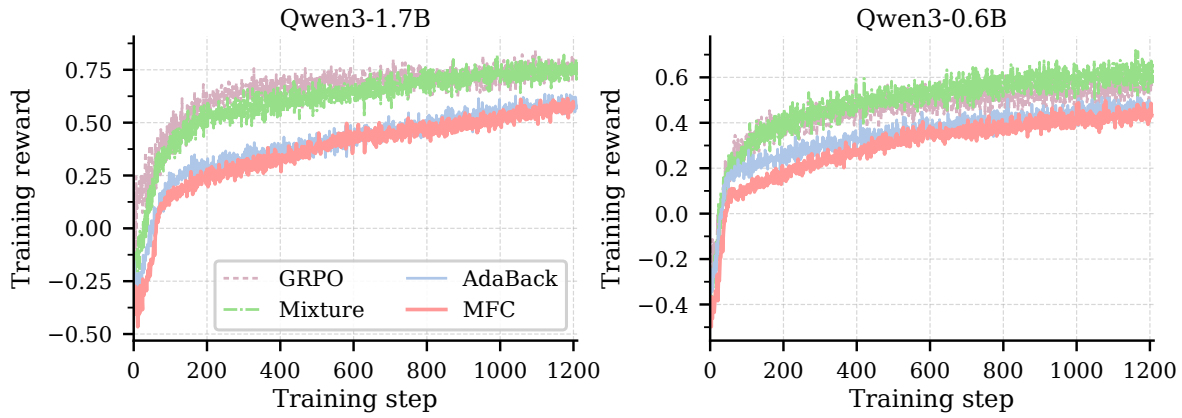


Figure 7: Training reward (mean per batch) across 1,200 steps for all four methods on both base models. Note that training reward is not directly comparable across methods because curriculum mechanisms present samples at different effective difficulty levels; see text for interpretation.

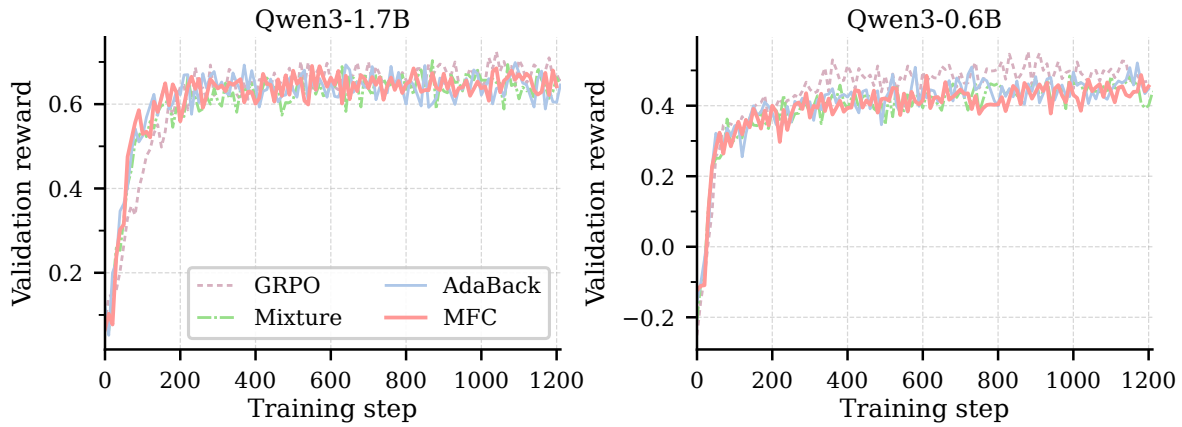


Figure 8: Validation reward on OR1-200 (no guidance, $\rho = 0$) evaluated every 10 steps. This is the fair comparison of unguided capability across methods. All methods converge by step 1,200; the curriculum methods (especially MFC) rise faster in the first ∼400 steps.

Decompose the curriculum gradient on the on-target ($g = 0$) / augmented ($g > 0$) partition,

$$\begin{aligned}\nabla J_{\mathcal{C}}^{(t)}(\theta) \;&= M_0^{(t)}\, \nabla J_{\text{zero}}(\theta) \\ &\quad + \left(1 - M_0^{(t)}\right) \nabla J_{\text{aug}}(\theta),\end{aligned} \tag{6}$$

where ∇J_{zero} is the gradient of the conditional curriculum objective restricted to visits with $g(s, \rho) = 0$, ∇J_{aug} the analogous quantity restricted to $g > 0$, and $M_0^{(t)}$ as in Eq. (4). Substituting Eq. (6) into Eq. (5),

$$\begin{aligned}\Delta J_0 \;&= \eta\, M_0^{(t)} \left\langle \nabla J_0,\, \nabla J_{\text{zero}} \right\rangle \\ &\quad + \eta \left(1 - M_0^{(t)}\right) \left\langle \nabla J_0,\, \nabla J_{\text{aug}} \right\rangle \\ &\quad + O(\eta^2).\end{aligned} \tag{7}$$

The on-target term under unbiased sampling. Under the assumption that the conditional distribution of s given $g(s, \rho) = 0$ matches $\mathcal{D}$, ∇J_{zero} is an unbiased estimator of ∇J_0 in expectation, and the first term of Eq. (7) satisfies

$$\mathbb{E}\left[\left\langle \nabla J_0,\, \nabla J_{\text{zero}} \right\rangle\right] \;=\; \|\nabla J_0\|^2 \;\geq\; 0,$$

contributing a definite-sign improvement to J_0 at an expected rate proportional to $M_0^{(t)}$. This condition holds for Mixture's $\rho = 0$ row and for the portion of AdaBack visits induced by its fixed p_{zero} valve. It does not hold in general for MFC, since its lattice-floor probability $1/(g_{\text{curr}}(s) + 1)$ depends on the per-sample frontier. For MFC, $M_0^{(t)}$ remains the weight on unguided visits with the test-time input format, but those visits follow a frontier-reweighted distribution over problems rather than $\mathcal{D}$.

Method	OR1-200	Uns-22	MATH-500	AMC23	AIME24	AIME25	AIME26	SciBench	GPQA-D	Avg.
SFT	54.5	13.6	83.4	61.9	26.7	26.7	19.2	36.0	36.5	39.8

Table 6: Per-benchmark pass@1 (%) for the SFT baseline on Qwen3-1.7B. Compare with Base (36.8 avg) and MFC (47.3 avg) in Table 2.

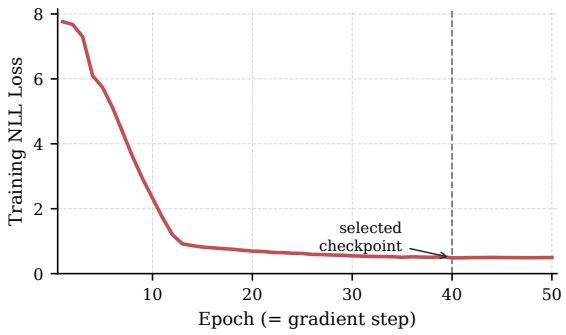


Figure 9: SFT training NLL loss over 50 epochs on Uns-128 (Qwen3-1.7B). The dashed line marks the selected checkpoint at step 40 where loss has effectively converged.

The augmented term has unconstrained sign. The second term of Eq. (7) is mediated entirely through parameter sharing across distinct prompts. Its sign and magnitude depend on whether the gradient on augmented prompts s_ρ is well-aligned with the gradient on the unguided prompt s, which is not in general guaranteed and can be:

- positive (helpful generalization, when reasoning learned under partial guidance transfers to unaided solving);
- zero (neutral, when the model learns hint-conditioned behaviors that do not transfer);
- negative (negative transfer, when the model overfits to teacher-style scaffolding or prompt artifacts that hurt unguided generation).

Cauchy–Schwarz bounds the magnitude by $\|\nabla J_{\text{aug}}\|\,\|\nabla J_0\|$, but does not pin the sign.

Implication for surrogate design. Eq. (7) suggests that a curriculum cannot expect to improve J_0 purely through augmented visits, but neither can it improve J_0 by visiting only $\rho = 0$ when those visits produce uniform failure (in which case $\|\nabla J_{\text{zero}}\|$ itself collapses by the same advantage-normalization mechanism that makes unsolvable problems inert under standard GRPO). Both factors of the on-target term, $M_0^{(t)}$ and $\|\nabla J_{\text{zero}}\|$, must be advanced jointly. MFC's per-sample frontier monotonically raises the probability of an unguided visit after a successful visit at a lower g (Algorithm 1). The value $g = 0$ remains in the sampling support from the first visit, although early unguided groups may still yield vanishing gradients before unaided success becomes likely.

Comparison with AdaBack on $M_0^{(t)}$. Conditional on the per-sample state, MFC's on-target visit probability is exactly $1/(g_{\text{curr}}(s)+1)$ by the discrete-uniform draw on $\{0, 1, \ldots, g_{\text{curr}}(s)\}$; since g_{curr} is monotone non-increasing under MFC's success-only update, this conditional probability is monotone non-decreasing across visits to s. AdaBack admits no such monotone bound: under the same discrete-uniform sampling (§3.1), its per-sample on-target visit probability is $p_{\text{zero}} + (1 - p_{\text{zero}})\,\mathbf{1}[g_{\min}(s) = 0] \,/\, (g_{\max}(s) - g_{\min}(s) + 1)$ because the valve overrides the uniform draw. When $g_{\min}(s) > 0$, the uniform-draw contribution vanishes and the on-target probability equals p_{zero}. A subsequent success at any g resets $g_{\min}(s)$ to 0, restoring uniform mass at $g = 0$. In our discrete unsolvable-only runs this mass is intermittent, and AdaBack's aggregate $M_0^{(t)}$ rises before plateauing around 0.40 (Figure 3).

G Prompt Templates

This section provides the prompt templates used for teacher trace generation and hint-based guidance throughout the paper.

G.1 Teacher Trace Generation

To obtain structured reasoning traces for each unsolvable problem, we prompt DeepSeek-V3.2 with the following system prompt. The output is required to consist of discrete `<step>` tags wrapped inside a `<solution>` block, followed by a single `<answer>` tag. We parse the output by extracting the content of each `<step>` tag into an ordered list of reasoning steps.

System Prompt for Teacher Trace Generation

[Persona Definition]
You are an expert mathematics teacher preparing educational material for students. Your goal is to explain the solution to a complex math problem in a way that is clear, logical, and easy to follow. Imagine your student is intelligent but may miss intermediate steps if you are not explicit.

[Strict Formatting Rules & Few-Shot Example]
Your entire response MUST be contained within a single <solution> tag. You MUST reason step by step. You MUST wrap each logical reasoning step in a <step>...</step> tag. You MUST wrap ONLY the final answer in an <answer>...</answer> tag. Do not include any additional explanations, comments, or text inside the <answer> tag. Do not output any text outside these tags.

Example:

User: The real roots of the equations $x^5 + x + 1 = 0$ and $x + \sqrt[5]{x} + 1 = 0$ are α, β respectively, then $\alpha + \beta$ equals?

Assistant:
<solution>
<step>Let's define a function based on the first equation: $f(x) = x^5 + x + 1$. Its derivative is $f'(x) = 5x^4 + 1$, which is always positive.</step>
<step>The first equation has real root α, so $f(\alpha) = 0$.</step>
<step>From the second equation, $\sqrt[5]{\beta} = -\beta - 1$.</step>
<step>Evaluating $f(-1-\beta) = (-1-\beta)^5 + (-1-\beta) + 1 = \beta - 1 - \beta + 1 = 0$.</step>
<step>Since f is strictly increasing, $\alpha = -1 - \beta$, so $\alpha + \beta = -1$.</step>
<answer>-1</answer>
</solution>

G.2 Hint-Based Guidance Prompt

During training, a guidance level ρ determines how many of the teacher's extracted steps are revealed to the student (Eq. 1). The selected steps are injected into the user message as a hint. The full prompt seen by the student model consists of a system message and a user message constructed as follows:

System Prompt (Student Model)

You are an expert mathematician with strong problem-solving skills. Think step by step.

User Message Template (with guidance)

{question}

Below are some initial reasoning steps that may help you:
{guidance_steps}

Please solve the problem step by step. You should use the provided steps as a reference, but do NOT just copy them. Instead, reconstruct the complete reasoning process in your own words, starting from the beginning, and continue the reasoning to find the final answer.

Use this format:
<think>
[Your reasoning process here]
</think>
\boxed{answer}

When $\rho = 0$ (no guidance), the hint block is omitted and the user message reduces to the question followed by the standard solving instruction.

H Broader Impacts

This work advances methods for improving mathematical reasoning in LLMs via reinforcement learning. The potential negative societal impacts align with those generally associated with LLM reasoning technologies: stronger reasoning capabilities could in principle be misused to generate more convincing misinformation or to automate harmful planning tasks. However, our contribution is methodological (a curriculum scheduling algorithm) and operates on publicly available base models; it does not introduce new capabilities beyond what larger-scale training on public data already enables. On the positive side, our data-efficiency finding ($\sim$16$\times$ reduction in required training data) may help democratize access to reasoning improvements for resource-constrained research groups.

I Licenses of Artifacts Used

Models. Qwen3-1.7B and Qwen3-0.6B (base models) are downloaded from their official HuggingFace repositories, where they are released under the Apache 2.0 license. DeepSeek-V3.2 (teacher model) was accessed exclusively through the OpenRouter API for the purpose of generating teacher reasoning traces; use of the API and the resulting outputs is governed by the provider's terms of service. We do not redistribute the model weights or any portion of them.

Training data. OpenR1-Math-220k is released under the Apache 2.0 license.

Evaluation benchmarks. MATH-500 is a subset of the MATH dataset; the source dataset and its splits are released under the MIT license. SciBench is released under the MIT license. GPQA-Diamond is a subset of GPQA, whose source dataset is released under CC-BY 4.0; the specific redistribution

we use does not specify a separate license. AMC23 and AIME24/25/26 are mathematics competition problems copyrighted by the Mathematical Association of America (MAA); we access them through community-redistributed datasets on HuggingFace and use them solely for non-commercial research evaluation, consistent with standard practice in the LLM reasoning literature.

Infrastructure. verl (training framework), vLLM (inference engine), and Math-Verify (answer verification) are all released under the Apache 2.0 license.

All artifacts are used in accordance with their stated terms. We release the training and in-domain evaluation datasets used in this work, together with the Uns-128 teacher traces, in a public Hugging Face dataset repository.[10] The repository documents the source of each record, the construction procedure, and the model used to generate the traces. The released bundle is distributed under the Apache 2.0 license, matching the Apache 2.0 license of its OpenR1-Math-220k source; we apply the same license to our own rights, if any, in the DeepSeek-V3.2 teacher traces, whose generation remains governed by the provider's terms of service. Because the upstream problems are drawn from olympiad, competition, and examination sources, a dataset-level license does not by itself clear rights in every underlying problem, and users remain responsible for those rights. We do not release model weights.

J Use of AI Assistants

Large language models (Claude, ChatGPT) were used to assist with grammar checking, paraphrasing, and improving clarity of the authors' original content.

[10] https://huggingface.co/datasets/yukangzhu/unlocking-the-unsolvable