LEARNING WHAT TO LEARN: LANGEVIN-BASED CURRICULUM OPTIMIZATION FOR RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards has substantially improved the reasoning capabilities of large language models, but its efficiency is often limited by uniform prompt sampling that ignores differences in training utility under the evolving policy. Curriculum learning offers a natural way to adapt training data to the evolving policy, yet existing methods either rely on predefined difficulty schedules or estimate utility at the individual-prompt level, where sparse rollout feedback can lead to unreliable estimates and self-reinforcing sampling bias. To address these challenges jointly, we propose **Langevin-based Adaptive Curriculum Optimization (LACO)**. LACO groups prompts into buckets based on their initial difficulty and pools online rollout feedback within each bucket, enabling more reliable estimation of their current training utility. It then uses Langevin Monte Carlo to adapt the sampling weights across buckets: utility-driven updates favor currently informative buckets, while stochastic perturbations preserve exploration of under-sampled buckets whose utility may change as the policy learns. This enables LACO to continuously track the policy's moving capability frontier while avoiding premature sampling concentration. Extensive experiments across multiple reasoning benchmarks demonstrate consistent improvements over uniform sampling and existing curriculum methods.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has substantially advanced the reasoning capabilities of large language models (Shao et al., 2024; Guo et al., 2025). However, most RLVR pipelines sample training prompts uniformly, despite substantial variation in their utility under the current policy. Prompts far below or beyond the policy's current capability often provide limited optimization signal (Yu et al., 2025; Foster et al., 2025), while prompts near the capability frontier tend to be more conducive to policy improvement (Bae et al., 2026; Gao et al., 2026). Consequently, uniform sampling can allocate substantial training computation to prompts of limited value, reducing the efficiency of RLVR (Zeng et al., 2026).

Curriculum learning provides an alternative to uniform sampling by allocating greater sampling probability to prompts that are more informative for the current policy. One line of works rely on predefined curricula, such as gradually progressing from easier to harder prompts (Parashar et al., 2026). However, a fixed curriculum schedule cannot adapt to the evolving competence of the policy, potentially resulting in persistent misalignment between the training distribution and the policy's capability frontier (Zhang et al., 2025). Another line of works adapt the curriculum online by estimating prompt-level utility and prioritizing prompts accordingly (Bae et al., 2026; Mao et al., 2026). Despite their adaptivity, such prompt-wise methods face two key limitations. First, the limited number of observations available for individual prompts leads to high-variance and unreliable utility estimates, particularly when training over large prompt pools. Second, adaptive sampling can progressively suppress prompts with low estimated utility, reducing their future observation frequency and thereby reinforcing their low sampling probabilities. This creates a **self-reinforcing sampling bias** that may prevent potentially useful prompts from being revisited as the policy evolves.

To address these limitations jointly, we propose **Langevin-based Adaptive Curriculum Optimization (LACO)**. Rather than estimating utility independently for each prompt, LACO groups prompts by initial difficulty and pools rollout feedback within each bucket, yielding more reliable utility esti-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

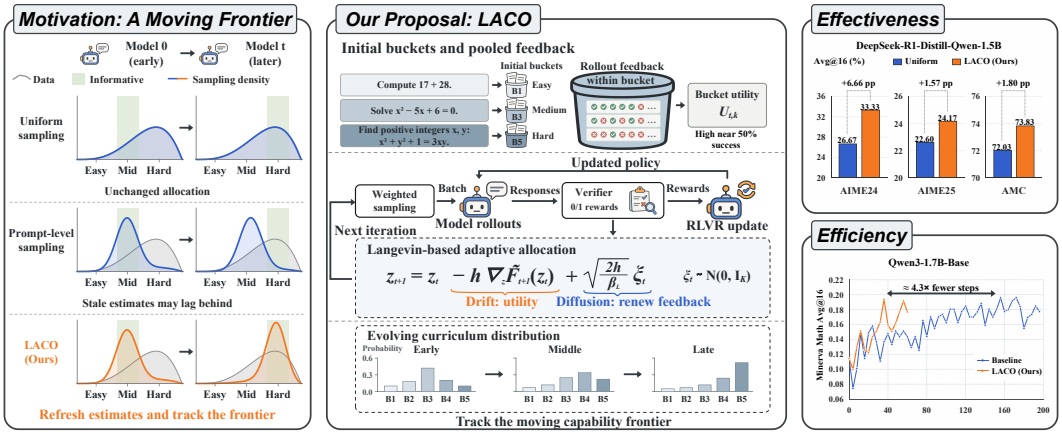


Figure 1: **Overview of LACO. Left (motivation):** As the policy improves, the informative region shifts, while sampling may remain unchanged or rely on stale estimates. **Center:** LACO pools feedback within initial-difficulty buckets and uses Langevin updates to adapt sampling weights, balancing utility-driven allocation with continued exploration. **Right:** Representative results show improved reasoning accuracy and training efficiency.

mates from sparse observations. LACO then adapts the bucket sampling weights through Langevin updates: utility-driven drift favors currently informative buckets, while stochastic diffusion allows under-sampled buckets to be revisited and their utility estimates refreshed as the policy evolves. Together, these mechanisms enable feedback-driven curriculum adaptation while mitigating premature sampling concentration. The drift term concentrates probability mass on regions with high estimated utility, while the diffusion term maintains exploration over under-observed regions that may become informative as the policy evolves. This drift–diffusion mechanism mitigates self-reinforcing sampling bias and enables the curriculum to continuously track the policy's moving capability frontier. Our contributions are summarized as follows:

- We introduce a bucket-level adaptive curriculum for RLVR that pools rollout feedback among prompts with similar initial difficulty, improving utility estimation under sparse observations.
- We propose LACO, which uses Langevin-based updates to adapt bucket sampling weights while revisiting under-sampled buckets and refreshing their utility estimates as the policy evolves.
- Across four models and multiple reasoning benchmarks, LACO achieves stronger overall performance and more efficient rollout utilization than uniform sampling and existing curriculum methods.

2 Related Work

Reinforcement Learning for LLM Post-Training. RL has become an important component of LLM post-training. Early work trains reward models from human preferences to improve text generation, summarization, and instruction following (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). PPO provides a widely used foundation for online policy optimization (Schulman et al., 2017). GRPO (Shao et al., 2024) estimates advantages from grouped responses without a separate critic , and DeepSeek-R1 demonstrates the potential of large-scale RL for complex reasoning (Guo et al., 2025). To improve training efficiency and stability, RLOO and REINFORCE++ explore critic-free policy optimization (Ahmadian et al., 2024; Hu et al., 2025), Dr. GRPO addresses optimization biases in GRPO (Liu et al., 2025). GSPO and SAPO further improve policy updates through sequence-level optimization and smooth gating, respectively (Zheng et al., 2025; Gao et al., 2025). LACO complements these advances by adapting the prompt-sampling distribution to improve the allocation of rollout computation during RLVR training.

Curriculum Learning and Data Selection. Curriculum learning improves training by adjusting the order or distribution of training data. Early work introduced easy-to-hard curricula (Bengio et al., 2009), while self-paced learning selects examples according to the model's current loss (Kumar et al., 2010). Automated curricula use learning progress to guide task selection through bandit algorithms or teacher policies (Graves et al., 2017; Matiisen et al., 2017).

In RLVR, curriculum design focuses on matching prompt difficulty to the evolving policy. E2H Reasoner gradually shifts training from easier to harder tasks through probabilistic scheduling (Parashar et al., 2026). DAPO filters prompt groups with identical rollout rewards (Yu et al., 2025), while Online Difficulty Filtering and LILO motivate prioritizing intermediate-difficulty or high-reward-variance prompts (Bae et al., 2026; Foster et al., 2025). To reduce the cost of filtering after generation, MoPPS estimates prompt success rates through streaming Bayesian inference and uses posterior sampling (Qu et al., 2026); DPS models evolving solving states with a hidden Markov model (Mao et al., 2026); and PCL trains a value model to identify prompts of intermediate difficulty (Gao et al., 2026). CurES derives an entropy-regularized prompt distribution and allocates rollout counts using Bayesian success-rate estimates (Zeng et al., 2026). LACO combines feedback pooling within fixed initial-difficulty buckets with Langevin updates of their mixture weights. Pooling provides shared utility estimates from sparse observations, while utility-driven drift and stochastic diffusion adapt the curriculum as the policy evolves.

3 PRELIMINARIES

We briefly review RLVR fine-tuning and adaptive curriculum learning. In RLVR, prompt sampling determines both the data used for policy optimization and the rollout feedback available for subsequent curriculum updates, creating a feedback loop between policy learning and data selection.

RLVR Fine-tuning and Group-Relative Learning Signals. RLVR fine-tuning optimizes a policy $\pi_\theta(y \mid x)$ using verifiable rewards. Given a prompt sampling distribution q over a training set $\mathcal{D}$, the objective is

$$\max_{\theta} \; \mathbb{E}_{x\sim q,\; y\sim\pi_\theta(\cdot|x)} \left[r(x, y) \right], \tag{1}$$

where $r(x, y) \in \{0, 1\}$ indicates whether the generated response y correctly solves prompt x.

Group-based RLVR methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) construct learning signals from multiple responses to the same prompt. Given a prompt x, the current policy generates G responses $\{y_{t,i}\}_{i=1}^{G}$ with rewards $r_{t,i} = r(x, y_{t,i})$, and the group-relative advantage is

$$\widehat{A}_{t,i} = \frac{r_{t,i} - \mathrm{mean}(\{r_{t,j}\}_{j=1}^{G})}{\mathrm{std}(\{r_{t,j}\}_{j=1}^{G}) + \epsilon}. \tag{2}$$

For binary rewards, groups containing both correct and incorrect responses yield nonzero reward-based advantages, whereas uniformly correct or incorrect groups provide no such signal. Thus, prompt utility depends on the current policy, with prompts of intermediate success probability typically providing more informative learning signals.

Adaptive Curriculum Learning in RLVR. We represent an adaptive curriculum by a prompt-sampling distribution q_t over $\mathcal{D}$, updated from rollout feedback as the policy evolves. Sampling from q_t determines not only which prompts are used for policy optimization, but also which prompts receive fresh feedback for future curriculum updates.

4 LANGEVIN-BASED ADAPTIVE CURRICULUM OPTIMIZATION

A key challenge in RLVR curriculum learning is the feedback loop between prompt selection and future curriculum updates: sampled prompts determine both the current policy update and the rollout feedback available for estimating utility. Prompt-level methods can therefore suffer from unreliable utility estimates under sparse feedback and from under-sampling prompts whose utility may change as the policy evolves. LACO addresses these issues with (i) bucket-level feedback aggregation for more reliable utility estimation and (ii) Langevin-based adaptive allocation that favors informative buckets while maintaining exploration of under-sampled ones.

4.1 Bucket-Level Utility Estimation under Sparse Feedback

Online rollout feedback is sparse at the individual-prompt level. For a training set of size N and batch size B, under independent sampling with replacement, the probability that a prompt receives no feedback for L consecutive steps is

$$\Pr\left(x_i \notin \bigcup_{s=t}^{t+L-1} \mathcal{B}_s\right) = \left(1 - \frac{1}{N}\right)^{BL} \approx \exp\left(-\frac{BL}{N}\right). \tag{3}$$

When N is large, prompt-level utility estimates can therefore become stale as the policy evolves. LACO addresses this by pooling rollout feedback within fixed difficulty buckets and adapting their sampling weights online.

Difficulty buckets and curriculum representation. We calibrate prompt difficulty using the initial policy π_{θ_0}. For each prompt x_i, we generate G_0 responses and estimate its initial success rate:

$$\hat{p}_i^{(0)} = \frac{1}{G_0} \sum_{j=1}^{G_0} r(x_i, y_{i,j}), \qquad y_{i,j} \sim \pi_{\theta_0}(\cdot \mid x_i). \tag{4}$$

We partition the success-rate range into K disjoint intervals $\{I_k\}_{k=1}^K$, defining fixed difficulty buckets

$$\mathcal{D}_k = \left\{x_i \in \mathcal{D} : \hat{p}_i^{(0)} \in I_k\right\}, \qquad k = 1, \ldots, K. \tag{5}$$

The bucket assignments remain fixed during training, while their sampling weights adapt. Let μ_k denote the uniform distribution over $\mathcal{D}_k$. LACO represents the curriculum as

$$q_t(x) = \sum_{k=1}^{K} w_{t,k} \mu_k(x), \qquad \boldsymbol{w}_t = (w_{t,1}, \ldots, w_{t,K}) \in \Delta_K. \tag{6}$$

Thus, curriculum adaptation reduces from N prompt-level probabilities to K bucket weights while allowing feedback to be shared among prompts of similar difficulty.

Online bucket-level utility estimation. At step t, let $\mathcal{B}_{t,k}$ denote the sampled prompt occurrences from bucket $\mathcal{D}_k$. We aggregate their rollout rewards to estimate the bucket success rate:

$$\bar{p}_{t,k} = \frac{1}{G|\mathcal{B}_{t,k}|} \sum_{x_i \in \mathcal{B}_{t,k}} \sum_{j=1}^{G} r(x_i, y_{i,j}), \qquad |\mathcal{B}_{t,k}| > 0. \tag{7}$$

We maintain an exponential moving average

$$\hat{p}_{t+1,k} = \alpha \hat{p}_{t,k} + (1 - \alpha)\bar{p}_{t,k}, \qquad \alpha \in [0, 1), \tag{8}$$

and, motivated by the group-relative learning signal in Section 3, define

$$U_{t+1,k} = -\left(\hat{p}_{t+1,k} - p^\star\right)^2, \qquad p^\star = 0.5. \tag{9}$$

Hence, buckets with intermediate current success rates receive higher utility.

Justification of bucket-level aggregation. Bucket-level estimation is effective when prompts grouped by initial difficulty retain similar success probabilities during training. Figure 2 shows that, although bucket-level success rates increase as the policy improves, prompts within each bucket remain relatively concentrated. This supports pooling feedback within fixed initial-difficulty buckets. [1]

[1] A theoretical analysis of within-bucket dispersion and its implications for bucket-level utility estimation is provided in Appendix C.

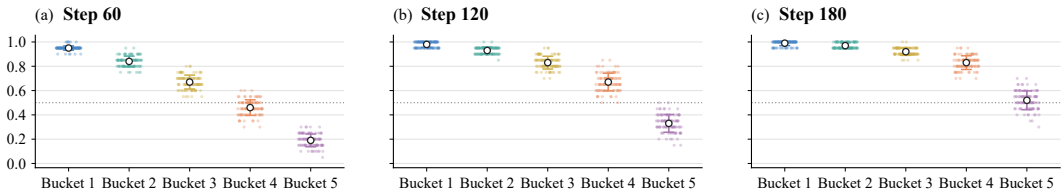


Figure 2: **Stability of difficulty buckets during training.** Prompts are grouped by initial difficulty. Although bucket-level success rates increase during training, within-bucket success rates remain relatively concentrated, supporting bucket-level utility estimation.

4.2 Langevin-Based Adaptive Curriculum Allocation

Given the bucket utilities $\boldsymbol{U}_{t+1} = (U_{t+1,1}, \ldots, U_{t+1,K})$ from Section 4.1, LACO adaptively updates the sampling weights $\boldsymbol{w}_t$ to favor informative buckets while avoiding premature concentration.

We define the potential

$$F_{t+1}(\boldsymbol{w}) = -\sum_{k=1}^{K} w_k U_{t+1,k} - \lambda H(\boldsymbol{w}), \qquad H(\boldsymbol{w}) = -\sum_{k=1}^{K} w_k \log w_k, \tag{10}$$

where the utility term favors informative buckets and the entropy term discourages excessive concentration. We parameterize $\boldsymbol{w}_t = \mathrm{softmax}(\boldsymbol{z}_t)$ with $\boldsymbol{z}_t \in \mathbb{R}^K$ and define

$$\widetilde{F}_{t+1}(\boldsymbol{z}) = F_{t+1}(\mathrm{softmax}(\boldsymbol{z})).$$

At each training iteration, LACO performs one Langevin Monte Carlo step:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - h\nabla_{\boldsymbol{z}} \widetilde{F}_{t+1}(\boldsymbol{z}_t) + \sqrt{\frac{2h}{\beta_L}}\boldsymbol{\xi}_t, \qquad \boldsymbol{\xi}_t \sim \mathcal{N}(0, I_K), \tag{11}$$

where $h > 0$ is the step size and $\beta_L > 0$ controls the diffusion strength. The next curriculum weights are $\boldsymbol{w}_{t+1} = \mathrm{softmax}(\boldsymbol{z}_{t+1})$.

Diffusion for feedback renewal. The deterministic drift favors high-utility buckets, while diffusion provides stochastic exploration that allows low-weight buckets to receive fresh feedback. To characterize this effect, let

$$\boldsymbol{m} = \boldsymbol{z}_t - h\nabla_{\boldsymbol{z}} \widetilde{F}_{t+1}(\boldsymbol{z}_t), \qquad \boldsymbol{v} = \mathrm{softmax}(\boldsymbol{m}).$$

After diffusion,

$$\boldsymbol{w}^+ = \mathrm{softmax}(\boldsymbol{m} + \sigma\boldsymbol{\xi}), \qquad \sigma^2 = \frac{2h}{\beta_L}, \quad \boldsymbol{\xi} \sim \mathcal{N}(0, I_K).$$

A local expansion yields

$$\mathbb{E}_{\boldsymbol{\xi}}[w_k^+] = v_k + \sigma^2 v_k\big(\|\boldsymbol{v}\|_2^2 - v_k\big) + \mathcal{O}(\sigma^4). \tag{12}$$

Thus, locally, diffusion increases the expected allocation of sufficiently low-weight buckets satisfying $v_k < \|\boldsymbol{v}\|_2^2$, creating opportunities to refresh their utility estimates as the policy evolves. Detailed derivations are provided in Appendix B.

4.3 Online RLVR Training Procedure

After the initial difficulty calibration, each training iteration samples prompts from q_t and generates G responses per prompt using π_{θ_t}. The same rollouts support both RLVR policy optimization and bucket-level utility estimation. The updated utilities drive the Langevin update (Equation 11), yielding the next curriculum distribution q_{t+1}. Algorithm 1 summarizes the complete procedure.

Table 1: **Main results on mathematical reasoning benchmarks.** All entries report Avg@16 (%). The best and second-best scores within each model block are shown in **bold** and underlined.

Base Model	Method	MATH-500	AIME'24	AIME'25	AMC	Minerva	Olympiad	Average
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	72.98	23.33	19.69	53.98	19.63	38.10	37.95
	Uniform	81.20	26.67	22.60	72.03	27.92	52.56	47.16
	MoPPS	82.23	26.94	22.67	72.50	28.47	52.74	47.59
	E2H	82.79	28.75	22.50	73.59	27.16	51.74	47.76
	LACO (Ours)	**83.72**	**33.33**	**24.17**	**73.83**	**28.52**	**54.09**	**49.61**
DeepSeek-R1-Distill-Qwen-7B	Initial Model	85.81	40.83	31.67	77.97	31.89	56.16	54.06
	Uniform	90.73	46.04	33.33	87.03	34.72	65.60	59.58
	MoPPS	91.24	48.54	34.58	87.19	35.42	65.60	60.43
	E2H	87.26	40.00	32.75	87.28	33.69	61.92	57.15
	LACO (Ours)	**91.26**	**52.29**	**37.71**	**89.22**	**36.47**	**65.97**	**62.15**
Qwen3-1.7B-Base	Initial Model	27.62	1.87	1.04	13.91	7.74	13.81	11.00
	Uniform	64.16	5.42	5.00	34.22	19.94	30.73	26.58
	MoPPS	64.45	6.67	5.00	35.31	21.19	31.15	27.30
	E2H	64.86	6.46	**6.04**	35.47	**23.18**	31.08	27.85
	LACO (Ours)	**66.00**	**9.17**	5.21	**36.72**	22.29	**32.00**	**28.57**
Qwen3-8B-Base	Initial Model	62.96	9.38	8.54	42.34	17.90	32.16	28.88
	Uniform	80.46	20.42	16.04	60.00	25.94	46.99	41.64
	MoPPS	79.54	**23.33**	17.92	62.66	25.21	48.30	42.83
	E2H	75.04	20.00	17.92	62.37	27.11	45.72	41.36
	LACO (Ours)	**82.59**	**23.33**	**21.67**	**65.47**	**28.93**	**48.67**	**45.11**

5 EXPERIMENTS

In this section, we conduct comprehensive experiments to evaluate LACO against representative baselines. Detailed descriptions of the datasets, models, implementation settings, evaluation protocols, and example instances are provided in Appendix E and F. Our experiments are organized around the following research questions: **RQ1:** Does LACO improve reasoning performance over baseline sampling strategies across different models and tasks? **RQ2:** How does LACO adapt its curriculum during training and improve training efficiency? **RQ3:** How do LACO's key components affect its performance? **RQ4**: How robust is LACO across different reasoning tasks, RL algorithms, and hyperparameter settings?

5.1 EXPERIMENTAL SETUP

Tasks. We evaluate LACO in two reasoning domains: mathematical reasoning using OpenRS (Dang & Ngo, 2026) and scientific reasoning using ARC (Clark et al., 2018), training separate models for each domain. To assess its generality across model scales and types, we conduct experiments with four language models ranging from 1.5B to 8B parameters, including both base and reasoning-distilled models. We use GRPO (Shao et al., 2024) as the default RL algorithm within the verl framework (Sheng et al., 2024) and further examine LACO's compatibility with PPO (Schulman et al., 2017) and REINFORCE++ (Hu et al., 2025).

Baselines. We select three baselines to represent distinct prompt-sampling strategies: (i) **Uniform Sampling**, the standard setting for RLVR, a non-adaptive strategy that assigns equal sampling probability to each training prompt; (ii) **E2H Reasoner** (Parashar et al., 2026), a predefined curriculum that progressively shifts sampling toward harder prompts; and (iii) **MoPPS** (Qu et al., 2026), an online adaptive strategy that uses Bayesian inference over individual prompt success rates and posterior sampling to guide selection. These baselines allow us to evaluate LACO against uniform allocation, predefined scheduling, and prompt-level online adaptation. Within each model setting, all methods share the same initial checkpoint, training data, and online rollout budget.

Benchmarks and Evaluation Metrics. We evaluate mathematical reasoning on six benchmarks: MATH-500 (Lightman et al., 2024), AIME 2024 and 2025 (Mathematical Association of America, n.d.a), AMC (Mathematical Association of America, n.d.b), Minerva Math (Lewkowycz et al.,

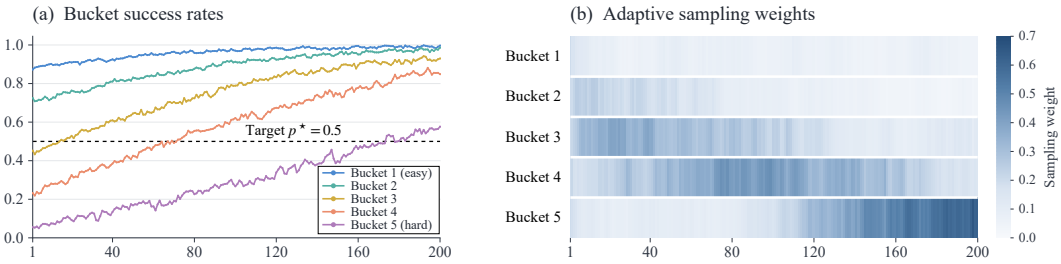


Figure 3: Curriculum adaptation over training steps. **(a)** Success rates for five buckets, ordered from easy to hard; the black dashed line marks the target $p^\star = 0.5$. **(b)** Corresponding sampling weights under LACO, with darker colors indicating greater sampling mass.

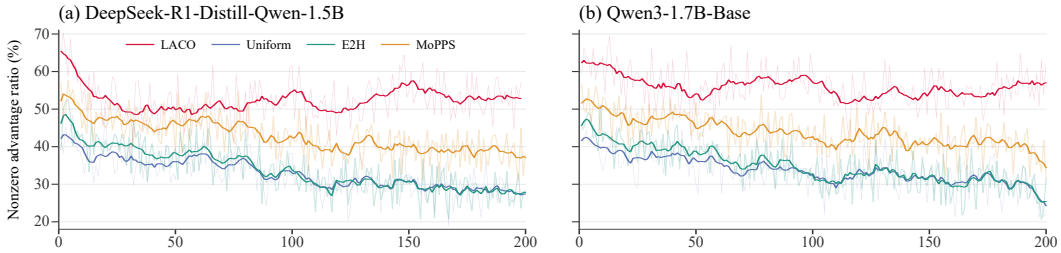


Figure 4: Nonzero advantage ratios.

2022), and OlympiadBench (He et al., 2024). We report Avg@16 (%), the mean accuracy over 16 sampled responses per question. For the ARC-trained model, we report Pass@1 accuracy (%) on PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), and MMLU (Hendrycks et al., 2021).

5.2 Main Results (RQ1)

Table 1 compares LACO with three baseline sampling strategies. LACO achieves the strongest overall performance, with particularly pronounced gains on challenging mathematical reasoning benchmarks. These results suggest that its advantage stems from a more effective allocation of the rollout budget across difficulty regions. Under imbalanced training distributions, baseline may devote substantial computation to prompts that have already been well learned. In contrast, LACO combines exploration of under-sampled regions with online utility estimation, progressively reallocating sampling mass toward regions that remain informative for the current policy. As the policy improves, bucket-level utilities are continuously updated, allowing the curriculum to shift toward newly informative, typically harder regions rather than repeatedly training on already mastered examples. This dynamic tracking of the policy's capability frontier provides a plausible explanation for LACO's stronger performance on challenging reasoning tasks. Although E2H outperforms LACO on two cases, its gains are less consistent across tasks, whereas LACO delivers more stable improvements overall, highlighting the robustness of adaptive bucket-level curriculum optimization.

5.3 Curriculum Adaptation and Rollout Utilization (RQ2)

To better understand the performance gains reported above, we analyze how LACO adapts its sampling distribution as the policy evolves and whether it increases the proportion of rollouts that provide informative learning signals.

Curriculum adaptation. Figure 3 illustrates how LACO tracks the policy's evolving capabilities and adjusts its sampling weights accordingly. By reallocating sampling mass in response to changes in real time, LACO keeps the training curriculum aligned with the model's current capabilities.

Sampling efficiency. To assess the efficiency of LACO's sampling strategy, we examine the proportion of rollout responses with nonzero reward-based advantages. Under GRPO, zero-advantage

responses consume generation computation without contributing to the reward-based policy-gradient objective. A higher nonzero-advantage ratio therefore indicates more effective use of the rollout budget, helping explain the observed training-efficiency gains. For each sampled prompt, we generate G responses with binary correctness rewards. We define the effective group ratio as $\eta_t = \frac{1}{B}\sum_{i=1}^{B} \mathbb{I}\left[0 < \sum_{j=1}^{G} r_{t,i,j} < G\right]$, where B is the number of sampled prompts. Under the group-relative advantage definition, groups containing both correct and incorrect responses yield nonzero reward-based advantages, whereas groups with identical rewards yield zero advantages. η_t equals the fraction of responses with nonzero reward-based advantages. Figure 4 compares LACO with three sampling strategies on DeepSeek-R1-Distill-Qwen-1.5B and Qwen3-1.7B-Base. Across both models, LACO consistently maintains the highest nonzero-advantage ratio throughout the observed training interval, indicating more efficient allocation of rollout computation to learning signals under the same online rollout budget.

5.4 Ablation Study (RQ3)

We examine the effects of LACO's components using Qwen3-1.7B-Base and the same settings.

Effects of online utility updates. Online utility updates allow LACO to adapt its curriculum to the policy's evolving capabilities. We freeze utilities at their initial calibration values while retaining sampling-weight updates, diffusion, and entropy regularization. As shown in Table 2, this causes the largest performance drop among the component ablations, which highlights the importance of refreshing utility estimates. As the policy improves, the learning value of different difficulty regions changes, making initial estimates increasingly outdated. Online updates provide allocation decisions with feedback that reflects the policy's current capabilities.

Table 2: **Ablation of LACO components.**

Variant	Avg@16 (%)	Δ (pp)
Uniform	26.58	−1.99
Fixed utilities	27.12	−1.45
w/o diffusion	27.99	−0.58
w/o entropy	28.40	−0.17
Full LACO	**28.57**	—

Effects of entropy regularization. We set $\lambda = 0$ while retaining online utility updates and diffusion. The resulting performance drop in Table 2 suggests that discouraging excessive sampling concentration remains beneficial even with stochastic exploration. Entropy regularization encourages broader allocation across difficulty regions, preserving opportunities to collect feedback beyond the currently preferred buckets.

Effects of diffusion. Removing Gaussian diffusion while retaining online utility updates and entropy regularization reduces average accuracy (Table 2). To examine the associated feedback behavior, we track the same initially low-weight bucket and measure how many training steps have passed on average, since each of its latest 32 prompt groups was generated. Lower values indicate that the observations were collected more recently. Figure 5 shows that the no-diffusion variant retains older observations. This supports the role of diffusion in renewing feedback from under-sampled buckets, providing opportunities to refresh utility estimates as the policy evolves.

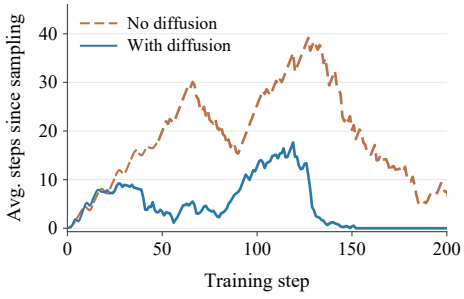


Figure 5: Average number of training steps elapsed since each prompt was sampled.

5.5 Robustness Analysis (RQ4)

General reasoning performance. To assess whether LACO generalizes beyond mathematical reasoning, we finetune Qwen3-1.7B-Base on ARC (Clark et al., 2018) with GRPO and evaluate the resulting policy on four general reasoning benchmarks: PIQA for physical commonsense reasoning, HellaSwag for commonsense completion, WinoGrande for commonsense and coreference reasoning, and MMLU for broad academic knowledge and reasoning. We compare LACO with uniform sampling under the same training configuration and rollout budget. As shown in Table 3, LACO im-

Table 3: General reasoning accuracy (%) of Qwen3-1.7B-Base finetuned on ARC.

Benchmark	Uniform	LACO
PIQA	77.53	**78.18**
HellaSwag	64.63	**66.32**
WinoGrande	52.41	**54.22**
MMLU	60.99	**61.33**

Table 4: Optimizer compatibility. R++ denotes REINFORCE++.

Method	MATH-500	AIME'25	Olympiad
Uniform-PPO	60.48	3.54	27.77
LACO-PPO	**61.49**	**3.96**	**29.00**
Uniform-R++	59.09	2.92	26.47
LACO-R++	**59.89**	**5.00**	**26.98**

proves performance consistently across all four benchmarks, suggesting that its benefit is not specific to mathematical reasoning but extends to more general reasoning domains. The largest gain appears on WinoGrande (+1.81 pp), followed by HellaSwag (+1.69 pp), indicating that adaptive curriculum allocation can particularly benefit tasks that require contextual and commonsense reasoning rather than relying solely on domain-specific knowledge.

Performance across RL algorithms. To examine whether LACO provides a general curriculum mechanism independent of the underlying RL optimizer, we further integrate it with PPO and REINFORCE++ when finetuning Qwen3-1.7B-Base on OpenRS. Table 4 compares LACO against uniform sampling within each optimizer on MATH-500, AIME 2025, and OlympiadBench. Clearly, LACO consistently improves performance across three benchmarks under both PPO and REINFORCE++, with gains of up to 1.23 points under PPO and 2.08 points under REINFORCE++. The largest improvement occurs on AIME'25 with REINFORCE++, where accuracy increases from 2.92 to 5.00. These results indicate that LACO is not tied to GRPO-specific updates, but instead provides a broadly compatible curriculum mechanism that improves rollout allocation across different reinforcement learning optimizers.

Effects of bucket granularity. To examine the sensitivity of LACO to the granularity of difficulty partitioning, we vary the number of buckets as $K \in \{2, 3, 4, 5, 10\}$ while keeping all other settings unchanged. The number of buckets controls both the resolution of difficulty partitioning and the amount of feedback pooled within each bucket. As shown in Figure 6, $K = 5$ performs best among the tested settings. This pattern is consistent with a trade-off between difficulty resolution and estimation reliability. Coarse partitions can pool prompts with different learning values, limiting targeted allocation. Finer partitions distribute limited feedback across more buckets, potentially reducing the reliability of utility estimates. A moderate granularity therefore provides sufficient differentiation across difficulty regions while preserving reliable utility estimation through feedback sharing within each bucket.

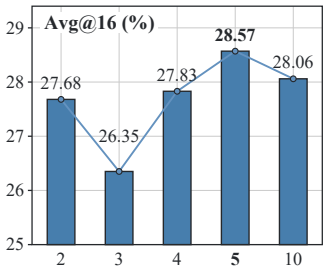


Figure 6: Sensitivity to the number of buckets K.

6 Conclusion

RLVR has emerged as an effective paradigm for improving the reasoning capabilities of large language models, yet its training efficiency remains constrained by how rollout computation is allocated across prompts of varying and evolving utility. To address this challenge, we introduced Langevin-based Adaptive Curriculum Optimization (LACO), which pools online rollout feedback within initial-difficulty buckets and adaptively updates their sampling weights through Langevin dynamics that balance utility-driven allocation with stochastic exploration. Across four models and six mathematical reasoning benchmarks, LACO improves average Avg@16 over uniform-sampling GRPO by 1.99–3.47 percentage points, while consistently outperforming existing curriculum baselines. These results demonstrate that effective RLVR curricula should adapt to the policy's evolving learning utility rather than follow a fixed or overly localized sampling strategy, and support treating prompt sampling as an online optimization problem that evolves jointly with the policy.

AI Assistance Disclosure

In this work, we used generative AI tools only to assist with editing and polishing the writing, including improving the clarity, readability, and presentation of the manuscript. We have not used generative AI tools for research ideation, development of theoretical models, formulation or verification of mathematical claims, design or execution of experiments, generation of synthetic data, or drafting sections of the paper.

All AI-assisted edits were carefully reviewed by the authors for accuracy and consistency with the intended meaning. We take full responsibility for the final content of this work, including all text, claims, and results presented in the manuscript.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024. URL https://arxiv.org/abs/2402.14740.

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. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.30. URL https://aclanthology.org/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. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Yanqi Dai, Yuxiang Ji, Xiao Zhang, Yong Wang, Xiangxiang Chu, and Zhiwu Lu. Harder is better: Boosting mathematical reasoning via difficulty-aware GRPO and multi-aspect question reformulation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2601.20614.

Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small LLMs: What works and what doesn't. In *Logical and Symbolic Reasoning in Language Models @ AAAI 2026*, 2026. URL https://openreview.net/forum?id=3pWL6Zxc4A.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. LILO: Learning to reason at the frontier of learnability. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1634. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/45d5823950cab9a88e10c6e8dbff351c-Abstract-Conference.html.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft Adaptive Policy Optimization. *arXiv preprint arXiv:2511.20347*, 2025. URL https://arxiv.org/abs/2511.20347.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient LLM post-training. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.01135.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1311–1320. PMLR, 2017. URL https://proceedings.mlr.press/v70/graves17a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025. URL https://arxiv.org/abs/2501.03262.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems*, volume 23, pp. 1189–1197, 2010. URL https://papers.nips.cc/paper_files/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

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. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*, 2025. URL https://arxiv.org/abs/2503.20783.

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Dynamics-predictive sampling for active RL finetuning of large reasoning models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2ae33575c3374050654ae7802326c81d-Abstract-Conference.html.

Mathematical Association of America. MAA invitational competitions. https://maa.org/maa-invitational-competitions/, n.d.a. American Invitational Mathematics Examination (AIME). Accessed September 24, 2026.

Mathematical Association of America. American mathematics competitions. https://maa.org/student-programs/amc/, n.d.b. Accessed September 24, 2026.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-Student Curriculum Learning. *arXiv preprint arXiv:1707.00183*, 2017. URL https://arxiv.org/abs/1707.00183.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

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. URL https://arxiv.org/abs/2506.06632.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating RL finetuning of reasoning models? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2026. doi: 10.1145/3770854.3780263. URL https://doi.org/10.1145/3770854.3780263.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd Schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/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. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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. URL https://arxiv.org/abs/2409.19256.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*, 2020. URL https://arxiv.org/abs/2009.01325.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. CurES: From gradient analysis to efficient curriculum learning for reasoning LLMs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.01037.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. 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*, pp. 6619–6633. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.336. URL https://aclanthology.org/2025.emnlp-main.336/.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593*, 2019. URL https://arxiv.org/abs/1909.08593.

A Why Intermediate-Difficulty Prompts Maximize Utility

Following the group-relative advantage analysis of Dai et al. (2026), we motivate the target success rate $p^\star = 1/2$ through the total absolute advantage within a rollout group.

Consider G responses to a prompt, with binary rewards $r_i \in \{0, 1\}$ and empirical success rate

$$\hat{p} = \frac{1}{G}\sum_{i=1}^{G} r_i. \tag{13}$$

For $0 < \hat{p} < 1$, the standard-deviation-normalized group-relative advantage, omitting the numerical stabilizer, is

$$\hat{A}_i = \frac{r_i - \hat{p}}{\sigma}, \qquad \sigma^2 = \frac{1}{G}\sum_{i=1}^{G}(r_i - \hat{p})^2. \tag{14}$$

The group contains $G\hat{p}$ correct responses and $G(1-\hat{p})$ incorrect responses. Therefore,

$$\sigma^2 = \hat{p}(1-\hat{p})^2 + (1-\hat{p})\hat{p}^2 = \hat{p}(1-\hat{p}). \tag{15}$$

Summing the absolute advantages of both types of responses gives

$$\begin{aligned} M(\hat{p}) &:= \sum_{i=1}^{G} |\hat{A}_i| \\ &= G\hat{p}\frac{1-\hat{p}}{\sqrt{\hat{p}(1-\hat{p})}} + G(1-\hat{p})\frac{\hat{p}}{\sqrt{\hat{p}(1-\hat{p})}} \\ &= 2G\sqrt{\hat{p}(1-\hat{p})}. \end{aligned} \tag{16}$$

Here, $M(\hat{p})$ measures the total magnitude of the advantage coefficients before clipping, rather than the norm of the parameter gradient. Groups with identical rewards are assigned zero advantages.

Since

$$\hat{p}(1-\hat{p}) = \frac{1}{4} - \left(\hat{p} - \frac{1}{2}\right)^2 \le \frac{1}{4}, \tag{17}$$

we obtain $M(\hat{p}) \le G$, with equality if and only if $\hat{p} = 1/2$. Thus, the total absolute advantage is maximized when half of the responses are correct and decreases as the success rate approaches either endpoint. When G is odd, the maximum occurs at the two attainable success rates nearest to $1/2$.

This result motivates the utility in equation 9. With $p^\star = 1/2$, it satisfies

$$U(p) = -\left(p - \frac{1}{2}\right)^2, \qquad 2G\sqrt{p(1-p)} = 2G\sqrt{\frac{1}{4} + U(p)}. \tag{18}$$

Because the right-hand side is increasing in $U(p)$, the utility ranks success rates in the same order as the total absolute group-relative advantage. LACO applies this criterion to estimated bucket success rates, favoring buckets near $p^\star = 1/2$.

B Diffusion, Expected Allocation, and Feedback Acquisition

B.1 Local Expansion of Expected Sampling Weights

We condition on the current feedback and the logits after deterministic drift, $\mathbf{m} \in \mathbb{R}^K$. Write $s_k(\mathbf{z}) = \mathrm{softmax}_k(\mathbf{z})$, $\mathbf{v} = \mathrm{softmax}(\mathbf{m})$, and $S_2 = \sum_j v_j^2$. The noisy weights are $w_k^+ = s_k(\mathbf{m} + \sigma\boldsymbol{\xi})$ with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$. All asymptotic expansions below hold for fixed $\mathbf{m}$ and K as $\sigma \to 0$; for the LACO update, this varies diffusion strength while holding the drift result fixed.

The first and second coordinate derivatives of softmax are

$$\partial_j s_k = s_k(\delta_{kj} - s_j), \tag{19}$$

$$\partial_{jj} s_k = s_k\big[(\delta_{kj} - s_j)^2 - s_j(1-s_j)\big], \tag{20}$$

where δ_{kj} is the Kronecker delta. Summing over coordinates at $\mathbf{m}$ yields

$$\Delta s_k(\mathbf{m}) = v_k\big[(1 - 2v_k + S_2) - (1 - S_2)\big] = 2v_k(S_2 - v_k). \tag{21}$$

Softmax has bounded derivatives through order four. Taylor expansion through order three therefore has a remainder bounded by $C\sigma^4\|\boldsymbol{\xi}\|^4$. The odd-order terms vanish in expectation, $\mathbb{E}[\xi_i\xi_j] = \delta_{ij}$, and the fourth Gaussian moment is finite. Consequently,

$$\mathbb{E}[w_k^+] = v_k + \frac{\sigma^2}{2}\Delta s_k(\mathbf{m}) + O(\sigma^4) \tag{22}$$

$$= v_k + \sigma^2 v_k(S_2 - v_k) + O(\sigma^4), \tag{23}$$

which establishes equation 12. If $v_k < S_2$, the coefficient of σ^2 is strictly positive, so $\mathbb{E}[w_k^+] > v_k$ for all sufficiently small positive σ. For a batch of B independent prompt draws conditional on $\mathbf{w}^+$, the expected region count is $B\mathbb{E}[w_k^+]$.

B.2 Probability of Obtaining New Feedback

Fix a positive integer batch size B and define $a_B(r) = 1 - (1 - r)^B$. Conditional on region weight r, this is the probability of observing the region at least once in the next batch. For $r = v_k \in (0, 1)$, the softmax derivatives satisfy

$$\|\nabla s_k(\mathbf{m})\|^2 = r^2(1 - 2r + S_2), \qquad \Delta s_k(\mathbf{m}) = 2r(S_2 - r). \tag{24}$$

Using the chain rule,

$$\Delta(a_B \circ s_k)(\mathbf{m}) = a_B'(r)\Delta s_k(\mathbf{m}) + a_B''(r)\|\nabla s_k(\mathbf{m})\|^2 \tag{25}$$

$$= B(1 - r)^{B-2} r\big[2S_2 - (B + 1)(1 + S_2)r + 2Br^2\big]. \tag{26}$$

The same Gaussian expansion, applied to the smooth function $a_B \circ s_k$, gives

$$\mathbb{E}[a_B(w_k^+)] - a_B(r) = \frac{\sigma^2}{2}\Delta(a_B \circ s_k)(\mathbf{m}) + O(\sigma^4). \tag{27}$$

If $r < S_2/(B + 1)$, then, since $S_2 \le 1$,

$$2S_2 - (B + 1)(1 + S_2)r + 2Br^2 > S_2(1 - S_2) + 2Br^2 > 0. \tag{28}$$

Thus the leading coefficient in equation 27 is positive, and

$$\mathbb{E}\big[1 - (1 - w_k^+)^B\big] > 1 - (1 - v_k)^B \tag{29}$$

for all sufficiently small positive σ. This provides a sufficient condition under which small diffusion increases the probability of obtaining new feedback from a bucket, supporting the feedback-renewal discussion in Section 4.2. In LACO, an observed bucket updates its success estimate according to equation 8, linking this visitation probability to an opportunity to refresh the utility estimate.

C Within-Bucket Dispersion and Utility Representativeness

Section 4.1 motivates pooling feedback within initial-difficulty buckets and examines their evolution in Figure 2. Here we give a quantitative interpretation of within-bucket concentration and explain how it supports using bucket-level success rates as utility proxies.

Bucket construction. The initial success-rate range $[0, 1]$ is partitioned into K disjoint intervals of equal width. The intervals are ordered from high to low success rates, so that bucket indices run from easy to hard. Bucket assignments remain fixed throughout training, allowing the same prompts to be tracked across training checkpoints.

A sufficient condition for bucket-mean coverage. Let $p_t(x)$ be the success probability of prompt x under the current policy, and let μ_k denote the uniform distribution over the prompts in a nonempty bucket k. For $x \sim \mu_k$, let $m_{t,k}$ and $\sigma_{t,k}$ denote the mean and standard deviation of $p_t(x)$, respectively:

$$m_{t,k} = \mathbb{E}_{x\sim\mu_k}[p_t(x)], \qquad \sigma_{t,k}^2 = \mathbb{E}_{x\sim\mu_k}\left[(p_t(x) - m_{t,k})^2\right].$$

For any tolerance $\delta > 0$, Chebyshev's inequality gives

$$\Pr_{x\sim\mu_k}(|p_t(x) - m_{t,k}| > \delta) \le \frac{\sigma_{t,k}^2}{\delta^2}.$$

Taking the complementary event yields

$$\Pr_{x\sim\mu_k}(|p_t(x) - m_{t,k}| \le \delta) \ge 1 - \frac{\sigma_{t,k}^2}{\delta^2}.$$

Consequently, for any coverage error $\gamma \in (0, 1)$,

$$\sigma_{t,k} \le \delta\sqrt{\gamma} \quad \Longrightarrow \quad \Pr_{x\sim\mu_k}(|p_t(x) - m_{t,k}| \le \delta) \ge 1 - \gamma. \tag{30}$$

Because μ_k is uniform over the bucket, this means that at least a $1 - \gamma$ fraction of its prompts have success probabilities within δ of the bucket mean. For $\delta = 0.1$ and $\gamma = 0.1$, the sufficient standard-deviation threshold is $\delta\sqrt{\gamma} \approx 0.0316$, corresponding to at least 90% of prompts having success probabilities within 0.1 of the mean.

Connection to the target success rate. Suppose that the bucket mean is close to the target success rate, $|m_{t,k} - 0.5| \le \varepsilon$ for some $\varepsilon \ge 0$. Whenever $|p_t(x) - m_{t,k}| \le \delta$, the triangle inequality gives

$$|p_t(x) - 0.5| \le |p_t(x) - m_{t,k}| + |m_{t,k} - 0.5| \le \delta + \varepsilon.$$

Consequently, under the sufficient condition in Equation 30,

$$\Pr_{x\sim\mu_k}(|p_t(x) - 0.5| \le \delta + \varepsilon) \ge 1 - \gamma. \tag{31}$$

Thus, a bucket mean within ε of 0.5, together with sufficiently low within-bucket dispersion, ensures that at least a $1 - \gamma$ fraction of its prompts have success probabilities within $\delta + \varepsilon$ of the target. This provides a rationale for the bucket-level utility proxy when these conditions hold. The statement concerns the current bucket mean $m_{t,k}$; using its online estimate $\hat{p}_{t,k}$ also depends on how accurately that estimate tracks the current mean.

D LACO Training Procedure

Algorithm 1 summarizes the complete LACO training procedure. We first calibrate prompt difficulty using the initial policy and construct fixed difficulty buckets. Sampling weights are initialized in proportion to bucket sizes, yielding uniform sampling over the prompt set. At each iteration, the same rollouts support both RLVR policy optimization and bucket-level success-rate estimation, with historical and current feedback combined through an exponential moving average. The resulting utility estimates guide a Langevin update of the sampling logits, whose softmax determines the curriculum distribution for the next iteration.

E Experimental Details

E.1 Datasets and Models

Training datasets. We conduct separate RLVR experiments on mathematical and scientific reasoning. For mathematical reasoning, we use OpenRS (Dang & Ngo, 2026), a curated collection of mathematical problems with reference answers. Its public release contains 7,000 problems, comprising 3,000 problems from Open-S1 and 4,000 from Open-DeepScaleR. For scientific reasoning, we use the AI2 Reasoning Challenge (ARC) (Clark et al., 2018), which consists of multiple-choice science questions drawn from grade-school examinations.

Algorithm 1 Langevin-based Adaptive Curriculum Optimization (LACO)

Require: Prompt set $\mathcal{D}$, initial policy π_{θ_0}, RLVR optimizer
Require: Number of buckets K, calibration responses G_0, prompt batch size B, rollout group size G, training steps T
Require: EMA coefficient α, entropy coefficient λ, step size h, inverse temperature β_L

Estimate $\hat{p}_i^{(0)}$ and construct $\{\mathcal{D}_k\}_{k=1}^K$ using Equations 4 and 5
Initialize $\hat{p}_{0,k} \leftarrow |\mathcal{D}_k|^{-1} \sum_{x_i \in \mathcal{D}_k} \hat{p}_i^{(0)}$ for all k
Set $w_{0,k} \leftarrow |\mathcal{D}_k|/N$ and $z_{0,k} \leftarrow \log w_{0,k}$ for all k, where $N = |\mathcal{D}|$
for $t = 0, \ldots, T-1$ **do**
    Sample B prompt occurrences from q_t to form $\mathcal{B}_t$
    Generate G responses per occurrence using π_{θ_t} and compute correctness rewards
    Update π_{θ_t} to $\pi_{\theta_{t+1}}$ with the RLVR optimizer using these responses and rewards
    Aggregate rewards for observed buckets using Equation 7
    Update $\hat{p}_{t+1,k}$ for all k using Equation 8
    Compute U_{t+1} using Equation 9
    Draw $\boldsymbol{\xi}_t \sim \mathcal{N}(0, I_K)$ and update $\boldsymbol{z}_{t+1}$ using Equation 11
    Set $\boldsymbol{w}_{t+1} \leftarrow \mathrm{softmax}(\boldsymbol{z}_{t+1})$
end for
return π_{θ_T}

Mathematical reasoning benchmarks. We evaluate mathematical reasoning on six benchmarks: MATH-500 (Lightman et al., 2024), AIME 2024 and AIME 2025 (Mathematical Association of America, n.d.a), AMC (Mathematical Association of America, n.d.b), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). MATH-500 contains 500 problems selected from the MATH test set. AIME and AMC contain competition mathematics problems, while Minerva Math and OlympiadBench extend the evaluation to quantitative reasoning and challenging olympiad-level mathematics, respectively.

General reasoning benchmarks. To assess transfer beyond the ARC training task, we evaluate the ARC-trained model on four benchmarks: PIQA (Bisk et al., 2020), which tests physical commonsense; HellaSwag (Zellers et al., 2019), which requires selecting plausible continuations of everyday situations; WinoGrande (Sakaguchi et al., 2020), which evaluates commonsense reasoning through contextual reference resolution; and MMLU (Hendrycks et al., 2021), which covers knowledge and problem solving across diverse academic and professional subjects.

Models. We consider four language models spanning 1.5B to 8B parameters: DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), together with Qwen3-1.7B-Base and Qwen3-8B-Base (Qwen Team, 2025). The DeepSeek models are Qwen-based models distilled using DeepSeek-R1-generated reasoning data, whereas the Qwen3 models are pretrained base checkpoints. This selection allows us to examine LACO across model scales and initial reasoning capabilities. All four models are used in the main mathematical reasoning experiments. We use Qwen3-1.7B-Base for the component ablations, bucket-granularity study, ARC-based general reasoning experiments, and optimizer-compatibility experiments. Within each comparison, all sampling methods start from the same initial model checkpoint.

E.2 Training Configurations

We implement our experiments using the verl framework (Sheng et al., 2024), with GRPO (Shao et al., 2024) as the default RL algorithm. Table 5 summarizes the training settings for the main mathematical reasoning experiments and the ARC experiment.

All experiments are executed on 8 NVIDIA A800 GPUs with 80GB memory.

Table 5: Training and LACO hyperparameters for GRPO.

Configuration	Value
Training and rollout	
Training steps T	200
Prompt batch size B	256
Responses per prompt G	8
Maximum prompt length	1024
Maximum response length	8192
Rollout temperature	1.0
Top-p	0.95
Correctness reward	$\{0, 1\}$
Policy optimization	
Actor learning rate	1×10^{-6}
Dynamic batching	Enabled
Group standard-deviation normalization	Enabled
LACO hyperparameters	
Number of difficulty buckets K	5
Target success rate $p^\star$	0.5
Curriculum entropy coefficient λ	0.01
Equivalent EMA coefficient α	0.99
Langevin step size h	0.01
Inverse temperature β_L	1.0
Sampling-weight update interval	1 step
Predictor success threshold	0.5
Sampler random seed	42
Implementation	
Rollout backend	vLLM
Number of GPUs	8
Rollout tensor parallel size	1
Gradient checkpointing	Enabled
Padding removal	Enabled

E.3 Evaluation Protocol

We perform evaluation using vLLM with a sampling temperature of 0.6 and top-p of 0.95. The maximum prompt and response lengths are set to 1,024 and 8,192 tokens, respectively. For mathematical reasoning, we sample 16 responses per question and report Avg@16 (%), defined as

$$\text{Avg@16} = \frac{100}{16N} \sum_{i=1}^{N} \sum_{j=1}^{16} \mathbf{1}[y_{i,j} \text{ is correct}] \,, \tag{32}$$

where N is the number of evaluation questions and $y_{i,j}$ is the j-th sampled response to question i. The Average column reports the arithmetic mean of Avg@16 across the six mathematical reasoning benchmarks.

F Data Examples

We provide two illustrative data examples for each reasoning domain. Each box presents the question, the instruction to reason step by step, the required final-answer format, and the reference answer.

F.1 Mathematical Reasoning

The following examples are drawn from the training split of OpenRS (Dang & Ngo, 2026).

OpenRS Data Example 1

Prompt:
The number 2^{1997} has m decimal digits, while the number 5^{1997} has n digits. Evaluate $m+n$.
Let's think step by step and output the final answer within $\texttt{\boxed}\{\}$.

Answer:
1998

OpenRS Data Example 2

Prompt:
Given that a is a multiple of 456, find the greatest common divisor of $3a^3 + a^2 + 4a + 57$ and a.
Let's think step by step and output the final answer within $\texttt{\boxed}\{\}$.

Answer:
57

F.2 General Reasoning

The following examples are drawn from the ARC-Challenge training split (Clark et al., 2018).

ARC Data Example 1

Prompt:
Which celestial object listed below has the greatest density?
(A) a planet
(B) a comet
(C) a nebula
(D) a neutron star
Let's think step by step and output the final answer within $\texttt{\boxed}\{\}$. The final answer must be a single option letter: A, B, C, or D.

Answer:
D

ARC Data Example 2

Prompt:
Puddles on a sidewalk are evaporating quickly. What most likely causes the puddles to evaporate?
(A) heat
(B) clouds
(C) air
(D) water
Let's think step by step and output the final answer within $\texttt{\boxed}\{\}$. The final answer must be a single option letter: A, B, C, or D.

Answer:
A