SAVE YOUR SATURATED DATA: LEARNING BEYOND REWARD SATURATION IN GROUP-BASED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative reinforcement learning (RL) relies on reward variation among sampled responses to estimate informative relative advantages. As language models become increasingly capable, existing training data can become *reward-saturated*: all sampled responses to the same problem might receive equally high rewards, where the group-relative learning signals vanish and leave previously useful data obsolete. In this work, we investigate *whether useful learning signals can be recovered from such saturated data*. We study interventions at four levels of group-policy RL pipelines—data, rollout, reward, and advantage—and conduct extensive RL training on saturated reasoning data *only*. While standard GRPO on saturated data would almost always yield near-0 advantages and near-noise signals, diverse interventions successfully recycle and repurpose such data: among the proposed strategies, interventions at rollout generation are consistently most effective: nudging the policy to generate "high-quality", incorrect solutions introduces rollouts with poor rewards into saturated groups as negative samples, which turns out to improve GRPO by 6.4% to 9.0% across Qwen3-1.7B and 4B. Other interventions such as increasing rollout temperature or adding auxiliary rewards can also restore non-zero advantages, but yield less consistent gains. Further analyses show that effective negative rollouts require informative negative trajectories, that the method remains effective alongside unsaturated data, and that it supports iterative recycling of newly saturated examples. While increasingly stronger LLMs would render more data as saturated, our results demonstrate that *don't waste your saturated data*: with the right strategies they can be recycled into useful RL training signals in an increasingly data-scarce world. Our code is available.[1]

1 INTRODUCTION

Group-based reinforcement learning (RL) has become a key component in LLM post-training, especially when used with verifiable rewards. Methods such as GRPO (Shao et al., 2024; DeepSeek-AI, 2025) and DAPO (Yu et al., 2025) sample multiple responses for a prompt, calculate rewards, and estimate advantage based on group reward differences. As a result, obtaining useful learning signals is contingent on having responses of diverse quality and reward values.

However, this learning signal vanishes at both extremes: When a problem is hard and all sampled rollouts are incorrect, uniformly low rewards won't leave meaningful advantage signals to train on. Prior work has primarily explored this problem through adaptive exploration (Zhang et al., 2026; Jiang et al., 2026; Agrawal et al., 2026), advantage modification (Le et al., 2026; He et al., 2026), and self-evolving curricula (Huang et al., 2026; Zhao et al., 2025). On the contrary, when a problem is easy and all rollouts are correct, uniformly high rewards suffer from the same advantage vanishing problem. We focus on the latter scenario, which we refer to as *reward saturation*, especially timely as LLM progress far exceeds the availability of challenging training data.

This reflects a growing tension between model capability and data utility: as the policy improves, more of its existing training data may become reward-saturated. While we can always curate harder problem sets and replace existing data, it is not sufficient alone: constructing high-quality challenging problems often requires substantial expert effort, reliable verification, and careful control over

[1] https://anonymous.4open.science/r/iclr2027-code-saturation-rl-8DCE/

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.

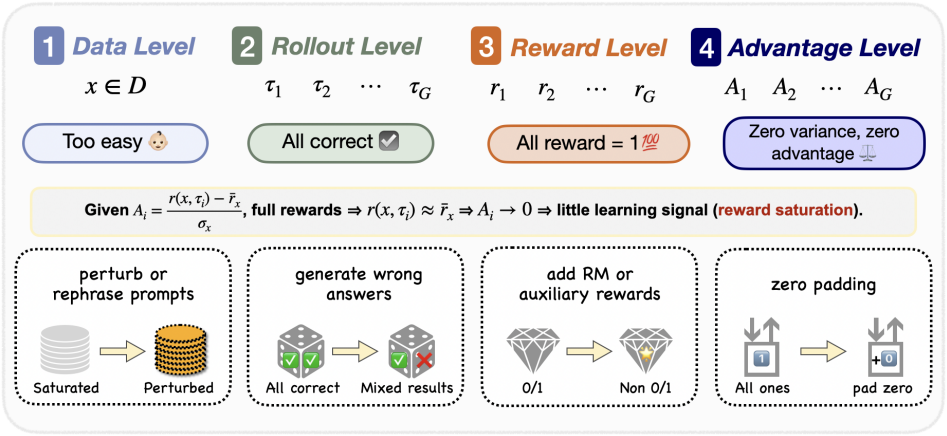


Figure 1: Overview of our work. (Top) Reward saturation in GRPO: When training data are easy, all G rollouts could be correct and receive the maximum reward, leading to zero advantage and thus little policy update. (Bottom) Multi-Level Interventions: We intervene at one of four stages of GRPO training: data, rollout, reward, and advantage to restore informative learning signals from reward-saturated examples. These interventions act at different stages of the training pipeline but share the same goal: restoring a usable group-relative learning signal for otherwise saturated prompts.

the problem distribution (Parashar et al., 2026; Huang et al., 2026; Bao et al., 2026). Our work asks a complementary question: *Can we recycle training data with saturated rewards for useful learning signals in group-based RL?*

While previous work (Liang et al., 2026) offered a preliminary step of intervening with rollout generation, we systematically investigate this research question through interventions at four levels of GRPO: *data* (e.g., adding irrelevant context to the training problem), *rollout* (e.g., constructing wrong negative rollouts), *reward* (e.g., adding additional fine-grained rewards), and *advantage* (e.g., modifying the advantage estimation function). We conduct extensive RL training across models of varying sizes and families, using these proposed interventions on reward-saturated data, and evaluate the trained policies across eight datasets spanning math, reasoning, and instruction following.

Across these interventions, we find that reward-saturated data can indeed be recycled into useful training signals. Among them, negative rollout is consistently the most effective: using the policy itself to generate incorrect solutions introduces informative trajectory-level contrasts within group, improving standard GRPO by 9.0% on Qwen3-1.7B and 6.4% on Qwen3-4B. Other interventions, like higher-temperature sampling, reward shaping, and advantage manipulation, can also recover learning signals, but yield smaller or less consistent gains. Further analyses show that effective negative rollouts require informative unsuccessful trajectories, remain effective when saturated and unsaturated data are mixed, enable new saturated examples to be iteratively recycled as the policy improves, and modifying the advantage function alone does not lead to better performance.

Taken together, our results suggest that reward saturation does not have to mark the end of a training example's utility in group-based RL. As models become stronger and increasingly more data become saturated, these saturated data can be continually recycled into useful RL training signals rather than discarded, enabling continued policy improvement even as saturation grows.

2 Related Work

Group-Based Reinforcement Learning with Verifiable Rewards. Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective paradigm for improving LLM reasoning. Group-based methods such as GRPO (Shao et al., 2024; DeepSeek-AI, 2025) and DAPO (Yu et al., 2025) estimate relative advantages from multiple rollouts of the same prompt, relying on within-group reward variation for learning. When all responses receive identical rewards, group-relative advantages collapse, motivating our work on intervention on four levels of policy training.

Recovering Learning Signals from Low-Information Groups. Prior work largely studies the difficult data regime, where most or all rollouts are incorrect. Approaches include filtering overly hard questions (Yu et al., 2025), improving exploration (Zhang et al., 2026; Jiang et al., 2026; Agrawal et al., 2026), modifying advantage estimation (Le et al., 2026; He et al., 2026), adapting the training distribution (Huang et al., 2026; Zhao et al., 2025), and constructing harder problems (Parashar et al., 2026; Huang et al., 2026; Bao et al., 2026). In contrast, we study the opposite: prompts with successful rollouts, where group-relative advantages likewise collapse.

Learning from Reward-Saturated Data. Recent work has begun to directly address reward saturation. Diagnostic studies show that advantage collapse strongly predicts training stagnation (He et al., 2026). Existing mitigations either avoid uninformative groups through pre-rollout difficulty estimation (Hu et al., 2026), reuse previously effective groups through off-policy replay (Mao et al., 2026), or recover within-group variation through virtual reward samples (He et al., 2026), auxiliary trajectory-quality signals (Deng et al., 2026), or constrained exploratory sampling (Liang et al., 2026). We extend this direction by systematically studying interventions across four stages of RL—*data*, *rollout*, *reward*, and *advantage*—to examine which recovered signals support effective learning, with our negative-rollout intervention making this distinction explicit.

3 METHOD

3.1 GRPO PRELIMINARY

GRPO (Shao et al., 2024) is a policy optimization algorithm that has demonstrated strong performance on various reasoning tasks. Let x denote a prompt sampled from dataset $\mathcal{D}$, and let $\{\tau_i\}_{i=1}^{G}$ denote a group of G rollouts generated by the current policy. Each rollout τ_i receives a binary verifiable reward $r(x, \tau_i) \in \{0, 1\}$, determined by whether its final answer is verified as correct against the ground-truth answer. GRPO then computes the group-level reward mean $\bar{r}_x$ and standard deviation σ_x to obtain the normalized group-relative advantage:

$$A(x, \tau_i) = \frac{r(x, \tau_i) - \bar{r}_x}{\sigma_x}. \tag{1}$$

For simplicity, we denote $A(x, \tau_i)$ as A_i. Let π_θ denote the optimized policy and π_{ref} denote a frozen reference policy. GRPO adopts a PPO-style clipped objective with token-level importance ratio $\rho_{i,t}(\theta)$ and KL regularization weight β:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\tau_i\sim\pi_{\theta_{\text{old}}}} \left[\sum_{t=1}^{|\tau_i|} \min\left(\rho_{i,t}(\theta)A_i, \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_i\right) \right] - \beta\mathbb{E}_{x\sim\mathcal{D}}\left[D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right]. \tag{2}$$

3.2 INTERVENTION AT DIFFERENT LEVELS

We define a rollout group as *reward-saturated* when all responses in the group receive the same maximum reward, such that the group-relative advantages vanish. In our setting, this corresponds to all G sampled responses being correct. As discussed in Eq. 2, GRPO relies on reward differences among multiple rollouts sampled from the same prompt to construct group-relative advantages. When all sampled trajectories receive identical rewards, the within-group reward variance collapses, resulting in uninformative relative advantages and limited policy updates.

To restore useful learning signals for such saturated groups, we investigate interventions at four stages of the GRPO pipeline: *data*, *rollout*, *reward*, and *advantage*. For each intervention, we first identify reward-saturated prompts by sampling G rollouts and retaining prompts for which all G responses are correct, and then apply the intervention to their saturated rollout groups. Although operating at different stages of training, all interventions share the same goal of restoring usable group-relative learning signals from saturated prompts. We next describe each intervention in turn.

Data-Level Intervention At the data level, we perturb the original training data to induce greater variation in rollout outcomes and thereby reduce reward saturation. We consider two variants:

- *Irrelevant Information:* augmenting each prompt with irrelevant information, yielding $D_{\text{irrelevant}} = \text{Augment}_{\text{irr}}(D)$.
- *Prompt Rephrasing:* rephrasing each prompt while preserving its underlying semantics, yielding $D_{\text{rewrite}} = \text{Rewrite}(D)$.

We investigate whether these data-level perturbations can reduce saturation by inducing greater variation in rollout outcomes and, consequently, within-group rewards.

Rollout-Level Intervention At the rollout level, we modify the generation process to increase reward heterogeneity within each rollout group. We consider two general strategies:

- *Higher Temperature:* increasing the sampling temperature t to encourage more diverse trajectories and increase the likelihood of obtaining both correct and incorrect rollouts.
- *Negative Rollout:* explicitly constructing incorrect rollouts for saturated groups in which all sampled responses receive the maximum verification reward.

For negative rollout construction, let $\mathcal{G}_{\text{correct}}$ and $\mathcal{G}_{\text{wrong}}$ denote the sets of correct and constructed incorrect rollouts, respectively. We form the rollout group as $\mathcal{G} = \mathcal{G}_{\text{correct}} \cup \mathcal{G}_{\text{wrong}}$.

To construct $\mathcal{G}_{\text{wrong}}$, we append an instruction to the original prompt that explicitly asks the current policy to generate an incorrect solution; the detailed prompt is provided in Appendix B.2. We further investigate alternative negative rollout construction strategies in Section 6.1.

Reward-Level Intervention At the reward level, we investigate whether alternative reward signals can alleviate saturation caused by coarse-grained verification rewards. Binary verification provides no distinction among trajectories once they all satisfy the correctness criterion. We therefore augment the verification reward with auxiliary signals that may differentiate otherwise equally rewarded trajectories. We consider three auxiliary reward signals, each evaluated independently:

- *Reward Model:* a reward-model score $R_{\text{RM}}(\tau_i)$ assigned to each rollout τ_i by a pretrained reward model.
- *Reasoning Quality:* an LLM-as-a-judge score $R_{\text{Reason}}(\tau_i)$ that evaluates the correctness of the reasoning process and intermediate steps in each rollout.
- *Response Diversity:* an LLM-as-a-judge score $R_{\text{Diversity}}(\tau_i)$ that measures the distinctiveness of each rollout's reasoning approach relative to the other $G - 1$ rollouts in the same group, assigning higher scores to more distinct trajectories.

For each intervention, we define the augmented reward as

$$R(\tau_i) = R_{\text{ver}}(\tau_i) + \lambda R_k(\tau_i), \qquad R_k \in R_{\text{RM}}, R_{\text{Reason}}, R_{\text{Diversity}}, \tag{3}$$

where R_{ver} denotes the original binary verification reward and λ controls the contribution of the auxiliary reward. Each R_k is evaluated separately rather than combining multiple auxiliary signals. These interventions test whether increasing reward granularity can restore informative group-relative advantages under reward saturation.

Advantage-Level Intervention One might argue that, since all of these rollouts achieve perfect reward, they should be treated equally with positive rewards. We investigate whether modifying advantage estimation can achieve this at the advantage level. We consider *Zero-Padding*, which appends k zero-reward entries to the original reward group when computing group-relative advantages. These additional entries increase the within-group reward variance, yielding non-zero advantages for the original successful rollouts.

Formally, given an original reward group $\{r(x, \tau_i)\}_{i=1}^{G}$, Zero-Padding augments it with k zeros and recomputes the group statistics:

$$A'(x, \tau_i) = \frac{r(x, \tau_i) - \bar{r}'_x}{\sigma'_x}, \quad \bar{r}'_x = \text{mean}\left(\{r(x, \tau_i)\}_{i=1}^{G} \cup \{\underbrace{0, \ldots, 0}_{k}\}\right), \tag{4}$$

where σ'_x denotes the standard deviation of the augmented reward group.

Zero-Padding restores non-zero relative advantages without modifying the training data, rollout trajectories, or reward function, while preserving the same positive preference across all rollouts. It therefore serves as a controlled intervention that increases the magnitude of the optimization signal without introducing new information.

4 EXPERIMENT

4.1 EXPERIMENT SETTINGS

We conduct RL training on a 4.3K subset of the MATH dataset (Hendrycks et al., 2021). To construct a reward-saturated setting, we retain only problems for which all eight independently sampled rollouts from the base model are correct. To evaluate both in-domain mathematical reasoning and broader generalization, we consider eight benchmarks spanning mathematical reasoning, general reasoning, and instruction following: MATH-500, Minerva (Lewkowycz et al., 2022), BBH (Suzgun et al., 2023), GPQA-Diamond (GPQA) (Rein et al., 2024), AIME24 (Veeraboina, 2023), AIME25, IFBench (Pyatkin et al., 2025), and IFEval (Zhou et al., 2023).

We use the Qwen3 series (Yang et al., 2025) as the primary backbone, including the 1.7B and 4B variants with the thinking mode disabled. All experiments are conducted on NVIDIA H200 GPUs using the *verl* framework (Sheng et al., 2025). Models are trained for two epochs with a rollout group size of $G = 8$. For negative rollout construction, we set $G_{\text{wrong}} = 2$ while keeping the total group size fixed. Additional implementation details are provided in Appendix B.

4.2 BASELINES

We compare our approach with representative baselines covering pretrained models, standard RL training, and prior methods for saturated RL data:

- **Base Model.** The pretrained backbone model without post-training.
- **GRPO.** The standard GRPO algorithm, serving as the primary RL baseline.
- **Mixed CUTS** (Liang et al., 2026). A baseline that constructs exploratory rollouts after warmup tokens by uniformly sampling to reduce saturation.
- **SFT.** Supervised fine-tuning on solutions generated by a teacher model (Qwen3-8B).
- **SFT $\rightarrow$ GRPO.** Supervised fine-tuning followed by standard GRPO training.

5 RESULTS

Table 1 reports results on saturated training data across two Qwen3 backbones.

Standard training baselines provide limited gains on saturated data. Standard GRPO yields only modest improvements over the base model, increasing the average score from 28.15 to 28.93 on Qwen3-1.7B and from 35.76 to 38.74 on Qwen3-4B. SFT alone provides little benefit over the base model, with average scores slightly decreasing from 28.15 to 27.51 on Qwen3-1.7B and from 35.76 to 35.62 on Qwen3-4B. Further applying GRPO after SFT reaches 27.62 and 37.61, respectively, but still falls short of standard GRPO (28.93 and 38.74). Mixed-CUTS is competitive on the smaller model, reaching 30.91, but drops below GRPO on Qwen3-4B (36.79). Overall, standard imitation-based training and existing strategies provide limited or inconsistent improvements when training data are already reward-saturated.

Negative rollouts are the most effective intervention under reward saturation. Among all evaluated strategies, negative rollout construction achieves the strongest overall performance on both model sizes. For Qwen3-1.7B, negative rollouts improve the average score from 28.93 with standard GRPO to 31.53, yielding a +2.60 absolute gain (+9.0% relative improvement) and outperforming Mixed-CUTS (30.91). For Qwen3-4B, the average score improves from 38.74 to 41.22, corresponding to a +2.48 absolute gain (+6.4% relative improvement). The gains are also pronounced on challenging mathematical reasoning benchmarks; for example, AIME25 improves from

Table 1: Performance comparison across eight benchmarks using Qwen3-1.7B and Qwen3-4B. We compare the base model, standard GRPO, Mixed-CUTS, SFT, SFT → GRPO and interventions at the data, rollout, reward, and advantage levels. We report avg@32 for AIME24 and AIME25 and pass@1 for all other benchmarks. The best and second-best results within each model group are highlighted in **bold** and underline, respectively.

Settings	MATH500	GPQA	BBH	Minerva	AIME24	AIME25	IFBench	IFEval	Average
Policy: Qwen3-1.7B									
Base Model	60.00	25.80	9.65	18.00	11.46	9.06	23.20	68.00	28.15
Baselines									
GRPO	62.60	24.24	10.86	17.28	17.08	9.17	21.20	69.00	28.93
Mixed-CUTS	61.00	**32.32**	10.55	18.38	16.77	12.81	25.20	**70.20**	30.91
SFT	58.40	29.29	8.23	18.01	6.67	10.10	22.00	67.40	27.51
SFT→GRPO	61.20	20.20	8.89	16.91	15.42	10.52	22.40	65.40	27.62
Data Level									
Irrelevant Information	62.20	20.71	10.04	16.18	16.56	7.50	22.40	68.80	28.05
Prompt Rephrasing	63.00	19.19	9.67	15.44	12.19	10.63	23.20	66.00	27.41
Rollout Level									
Higher temperature	63.60	25.25	10.40	17.28	16.25	12.71	24.00	68.40	29.74
Negative Rollout	64.80	29.80	**13.01**	**18.38**	**20.31**	**14.38**	24.00	67.60	**31.53**
Reward Level									
Reward Model	62.00	28.28	10.29	**18.38**	18.85	9.38	22.00	67.80	29.62
Reasoning Quality	**65.80**	27.27	10.40	16.54	16.88	10.42	23.60	69.20	30.01
Response Diversity	61.20	24.24	10.33	16.54	10.31	10.52	22.80	67.80	27.97
Advantage Level									
Zero Padding ($k = 2$)	62.80	27.27	10.84	16.54	16.04	10.52	22.40	66.80	29.15
Zero Padding ($k = 3$)	64.00	28.28	10.93	16.54	18.33	12.50	**25.60**	69.40	30.70
Policy: Qwen3-4B									
Base Model	67.20	38.89	15.00	23.90	19.58	13.13	28.40	80.00	35.76
Baselines									
GRPO	70.60	31.31	16.13	25.74	31.98	**27.40**	28.00	78.80	38.74
Mixed-CUTS	70.40	38.89	16.28	24.26	23.33	10.73	29.60	80.80	36.79
SFT	67.20	38.89	13.69	23.90	16.04	14.06	30.00	81.20	35.62
SFT→GRPO	69.40	31.31	14.93	25.37	30.21	19.69	27.20	82.80	37.61
Data Level									
Irrelevant Information	**71.00**	40.40	15.80	25.37	27.92	25.21	28.00	80.40	39.26
Prompt Rephrasing	68.60	34.34	15.93	24.63	22.08	23.23	29.60	80.40	37.35
Rollout Level									
Higher temperature	69.40	38.38	16.31	**26.10**	23.85	18.13	29.60	**82.60**	38.05
Negative Rollout	70.60	40.40	**18.27**	**26.10**	33.02	25.52	**33.20**	**82.60**	**41.22**
Reward Level									
Reward Model	70.20	**41.92**	17.83	22.79	**36.46**	20.73	31.20	80.80	40.24
Reasoning Quality	**71.00**	32.32	15.75	25.74	23.02	27.19	30.80	80.00	38.23
Response Diversity	68.20	34.34	15.29	23.53	13.44	12.92	31.20	82.00	35.11
Advantage Level									
Zero Padding ($k = 2$)	62.80	31.82	13.83	19.85	14.06	11.88	27.60	81.40	32.90
Zero Padding ($k = 3$)	67.20	30.30	13.50	22.79	20.21	23.13	30.00	79.80	35.87

9.17 to 14.38 on Qwen3-1.7B. These results show that reward-saturated examples need not be discarded: constructing informative negative trajectories introduces meaningful within-group reward contrast and restores useful learning signals for GRPO.

Increasing diversity alone provides limited and inconsistent improvements. Interventions that perturb the input or increase rollout diversity provide smaller and less consistent gains. Increasing the rollout temperature improves the average score from 28.93 to 29.74 on Qwen3-1.7B, but decreases it from 38.74 to 38.05 on Qwen3-4B. Data-level perturbations show a similar pattern: irrelevant information improves Qwen3-4B to 39.26 but provides no gain on Qwen3-1.7B, while prompt rephrasing underperforms GRPO at both scales. These results suggest that increasing diversity alone does not consistently overcome reward saturation; what matters is whether the generated trajectories introduce informative reward differences within each rollout group.

Reward shaping and advantage manipulation recover weaker signals. Reward-level and advantage-level interventions can partially restore learning signals, but remain less consistent than negative rollouts. Reward-model supervision is particularly effective on Qwen3-4B, improving the average score from 38.74 to 40.24, although it remains below negative rollouts at 41.22. In contrast, reasoning-quality and response-diversity evaluators do not consistently improve over standard GRPO. Zero-reward padding improves Qwen3-1.7B, reaching 30.70 with three additional zero-reward samples, but substantially underperforms GRPO on Qwen3-4B. Together, these results suggest that simply introducing non-zero advantages or finer-grained reward signals can help, but is less reliable than constructing meaningful trajectory-level contrasts.

6 Analysis

6.1 Effective Negative Rollouts Require Informative Trajectories

We investigate whether the benefit of negative rollouts arises simply from restoring reward variance, or whether the construction of the negative trajectories also matters. We compare three strategies that differ in how coherently the negative outcome is reflected throughout the trajectory: (1) *answer replacement*, which replaces only the final answer of an originally correct rollout while preserving its reasoning trajectory; (2) *wrong-answer-conditioned continuation*, which provides a reasoning prefix and conditions the model to continue toward a predetermined incorrect answer; and (3) *model-generated negative rollout*, our default strategy, which instructs the current policy to generate an incorrect solution from scratch. All three strategies restore reward variation within saturated groups, but differ in how faithfully the resulting trajectories represent unsuccessful generation.

As shown in Table 2, simply restoring reward variation does not consistently improve performance. Answer replacement performs worse than standard GRPO, with average scores of 27.70 on Qwen3-1.7B and 33.38 on Qwen3-4B. Wrong-answer-conditioned continuation improves over answer replacement, reaching 30.29 and 37.31, respectively, but remains below GRPO on Qwen3-4B. In contrast, model-generated negative rollouts achieve the strongest average performance on both backbones, reaching 31.53 on Qwen3-1.7B and 41.22 on Qwen3-4B.

These results show that introducing negative rollouts alone is not sufficient: *how the corresponding negative trajectories are constructed matters*. Answer replacement creates a mismatch between an otherwise successful reasoning trajectory and an artificially incorrect final answer, while conditioned continuation constrains only part of the generation process. In contrast, model-generated negative rollouts allow the policy to construct an entire unsuccessful trajectory, producing a more coherent correspondence between the generated trajectory and its negative reward.

Table 2: Comparison of negative rollout construction strategies across eight benchmarks. Although all three strategies restore reward variation within saturated groups, model-generated negative rollouts achieve the strongest average performance across both model scales.

Method	MATH500	GPQA	BBH	Minerva	AIME24	AIME25	IFBench	IFEval	Average
Qwen3-1.7B									
GRPO	62.60	24.24	10.86	17.28	17.08	9.17	21.20	69.00	28.93
Answer Replacement	59.20	27.78	13.10	14.71	2.50	7.29	29.20	67.80	27.70
Wrong-Answer-Cond.	64.20	29.80	10.86	17.28	17.19	9.17	23.20	70.60	30.29
Model-Generated Negative	64.80	29.80	13.01	18.38	20.31	14.38	24.00	67.60	**31.53**
Qwen3-4B									
GRPO	70.60	31.31	16.13	25.74	31.98	27.40	28.00	78.80	38.74
Answer Replacement	54.20	39.90	16.37	23.16	15.42	3.96	34.80	79.20	33.38
Wrong-Answer-Cond.	71.20	36.36	16.86	25.37	24.48	13.65	32.00	78.60	37.31
Model-Generated Negative	70.60	40.40	18.27	26.10	33.02	25.52	33.20	82.60	**41.22**

6.2 Negative Rollouts Generalize Beyond Fully Saturated Data

Our main experiments focus on fully saturated training data, while saturated and non-saturated groups may coexist in practice. We therefore evaluate Negative Rollout under different saturation

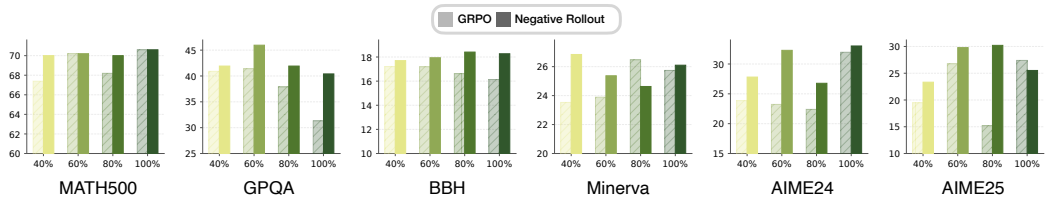


Figure 2: Performance of GRPO and Negative Rollout across different ratios of reward-saturated training data. Negative Rollout improves aggregate performance across all saturation ratios, with gains varying across benchmarks.

ratios. Before training, we identify saturated prompts and partition the data into saturated and non-saturated pools, from which we construct training sets with varying proportions. Negative Rollout is applied only to saturated groups, while non-saturated groups follow standard GRPO. This setting allows us to evaluate whether the method remains effective when reward saturation affects only part of the training data.

As shown in Figure 2, Negative Rollout improves the average performance over GRPO across all tested saturation ratios, with gains of 1.81–3.09 points (4.8–8.3% relative). The improvements span multiple reasoning benchmarks, although their magnitude varies across tasks and saturation ratios. For example, at 80% saturation, Negative Rollout substantially improves AIME 2025 (15.21→30.21) and GPQA (37.88→41.92), while some individual settings exhibit smaller gains or slight regressions. Importantly, the aggregate benefit persists even when only 40% of the training data is saturated, where average performance improves from 37.97 to 39.78. These results suggest that Negative Rollout does not require the entire training set to be reward-saturated. Instead, it can be selectively integrated into standard GRPO training: non-saturated groups retain their original relative learning signals, while Negative Rollout restores learning signals only for saturated groups. This makes the intervention applicable to mixed training settings in which saturation occurs only for a subset of prompts.

6.3 Towards RSI-Style Self-Evolving Saturation-Aware Training

Reward saturation is not necessarily a static property of a dataset: as the policy improves during RL training, previously challenging prompts may become saturated. To capture this dynamic, we periodically identify newly saturated prompts from the remaining data and add them to the saturation pool. As shown in Figure 3, the saturated pool expands by 443 newly discovered prompts by step 80, showing that new reward-saturated prompts continue to emerge as training progresses.

This iterative process is accompanied by continued performance gains, with validation accuracy on AIME 2025 increasing from 0.2208 at step 0 to 0.3271 at step 80. Rather than discarding these solved examples, negative rollout construction converts them back into prompts with informative learning signals. We refer to this iterative procedure as an RSI-style self-evolving saturation-aware training loop in which policy improvement creates newly saturated prompts, which are identified and converted into informative training signals to further improve the policy. Although our experiment is a preliminary realization of this paradigm, it suggests that training data can evolve together with model capability, reducing the need to continually replace solved examples with newly constructed harder problems.

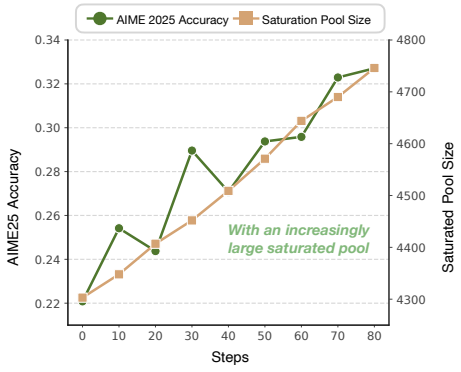


Figure 3: Self-evolving saturation-aware training, with expanding saturated pool and improving AIME25 performance.

6.4 A Larger Advantage Doesn't Lead to Better Performance

Zero-reward padding provides a direct way to restore non-zero group-relative advantages in fully saturated groups. Consider a saturated group of G successful rollouts with reward 1. After append-

ing k artificial samples with reward 0, the advantages for the original correct $A_+(k)$ and appended wrong $A_-(k)$ rollouts become:

$$A_+(k) = \sqrt{\frac{k(G+k-1)}{G(G+k)}}, \quad A_-(k) = -\sqrt{\frac{G(G+k-1)}{k(G+k)}}$$

Notably, $A_+(k)$ increases monotonically with k (full derivation in Appendix E). However, all G successful rollouts receive the same $A_+(k)$ regardless of their reasoning quality or trajectory structure. Increasing k therefore amplifies the advantage magnitude without providing additional distinctions among successful trajectories.

We empirically examine this effect on Qwen3-4B with $k \in \{1, 2, 3, 8, \text{rand}\}$. As shown in Table 3, performance does not improve monotonically with k, despite the monotonic increase in $A_+(k)$, and all zero-padding variants underperform standard GRPO. This reveals an important distinction between *advantage magnitude* and *advantage informativeness*: zero-reward padding restores non-zero advantages, but the resulting contrast is purely numerical and provides no additional trajectory-level information. Effective recovery from reward saturation therefore requires not only a non-zero learning signal, but also one that meaningfully distinguishes sampled trajectories.

Table 3: Effect of number of zero-reward padding size k on Qwen3-4B.

Settings	MATH500	GPQA	BBH	Minerva	AIME24	AIME25	IFBench	IFEval	Average
GRPO	70.60	31.31	16.13	25.74	31.98	27.40	28.00	78.80	38.74
Zero Padding ($k = 1$)	68.20	34.85	15.11	23.90	21.67	19.27	29.20	80.60	36.60
Zero Padding ($k = 2$)	62.80	31.82	13.83	19.85	14.06	11.88	27.60	81.40	32.90
Zero Padding ($k = 3$)	67.20	30.30	13.50	22.79	20.21	23.13	30.00	79.80	35.87
Zero Padding ($k = 8$)	62.20	29.80	13.50	22.79	17.71	3.65	29.60	79.60	32.36
Zero Padding (random k)	66.40	30.30	13.27	22.43	17.08	18.96	27.60	80.00	34.51

6.5 Generalization Across Model Families

To examine whether negative rollout construction generalizes beyond Qwen3, we further evaluate it on LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024). As shown in Table 4, GRPO improves the average score from 22.98 to 24.97, while negative rollout further improves it to 25.63, with pronounced gains on Minerva (13.97→20.59), GPQA (20.71→24.75), and AIME24 (3.30→10.00). In contrast, Mixed-CUTS transfers poorly to LLaMA-3.1-8B-Instruct, reducing the average score to 17.35. These results suggest that negative rollouts provide a more robust intervention for reward saturation across model families without model-specific adaptation.

Table 4: Generalization results on LLaMA-3.1-8B-Instruct across eight benchmarks.

Settings	MATH500	GPQA	BBH	Minerva	AIME24	AIME25	IFBench	IFEval	Average
Base Model	34.00	18.18	8.96	12.50	6.25	1.15	30.80	72.00	22.98
GRPO	45.00	20.71	10.42	13.97	3.30	1.04	32.90	72.40	24.97
Mixed-CUTS	27.00	3.03	2.50	9.93	3.30	0.21	29.20	63.60	17.35
Negative Rollout	43.60	24.75	9.27	20.59	10.00	0.42	29.20	67.20	**25.63**

7 Conclusion

We study reward saturation in GRPO, where all rollouts for a prompt receive the same high reward and therefore provide little relative advantage signal. Across interventions at the data, rollout, reward, and advantage levels, our experiments show that constructing wrong-answer rollouts at the rollout level is the most effective way to recover useful learning signals from saturated examples. The method consistently improves average performance across Qwen3-1.7B and Qwen3-4B, and transfers to LLaMA3.1-8B-Instruct. Our analyses further show that simply changing reward statistics, adding generic perturbations, or imitating teacher solutions is less reliable. Overall, the results suggest that saturated data can remain useful for RL when it is paired with semantically meaningful negative trajectories that reveal plausible reasoning failures.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing writing. We did not use generative AI tools to design or provide feedback on research methodology, conduct experiments, or generate synthetic datasets, and the remaining disclosure categories are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies reinforcement learning with reward-saturated data for large language models. Our experiments use public mathematical and general reasoning benchmarks and do not involve human subjects or private data. The negative trajectories introduced by our method are model-generated and used solely as training signals.

REPRODUCIBILITY STATEMENT

We provide extensive experiment details such as hyperparameter settings, dataset statistics, and more in Appendix B. The training and inference code is in `https://anonymous.4open.science/r/iclr2027-code-saturation-rl-8DCE/`

REFERENCES

Priyank Agrawal, Ankur Samanta, Shervin Ghasemlou, Jalaj Bhandari, Kavosh Asadi, Daniel Jiang, and Aditya Modi. Off-context grpo: Learning to reason on hard problems using privileged information, 2026. URL https://arxiv.org/abs/2607.19313.

Longtian Bao, Jianyou Wang, Yang Zhang, Youze Zheng, and Ramamohan Paturi. Question begets question: Self-evolving curriculum for reinforcement fine-tuning on competition mathematics, 2026. URL https://arxiv.org/abs/2608.01522.

Dejian Yang Haowei Zhang Jun-Mei Song et al DeepSeek-AI, Daya Guo. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Zeyun Deng, Yuzhe Lu, Yawei Wang, Linbo Liu, Qing Ping, Han Ding, Guande Wu, Panpan Xu, and Jun Huan. Prism-grpo: Faster vla policy optimization via splitting same-outcome groups, 2026. URL https://arxiv.org/abs/2608.17423.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,

Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ra-

maswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=MKNimf9bIx.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Zengjie Hu, Jiantao Qiu, Tianyi Bai, Haojin Yang, Binhang Yuan, Qi Jing, Conghui He, and Wentao Zhang. Vade: Variance-aware dynamic sampling via online sample-level difficulty estimation for multimodal reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9846–9855, June 2026.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning LLM from zero data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=96apU6YzSO.

Heyang Jiang, Henry Liu, and Baharan Mirzasoleiman. Learning as reasoning unfolds: Progressive rollout allocation for efficient reinforcement learning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=qoJJK90DGh.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Dac Lai, and Eunho Yang. 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*, 2026. URL https://openreview.net/forum?id=kiXFIESZKv.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh 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 *The Thirty-Sixth Annual Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Zhenwen Liang, Yujun Zhou, Sidi Lu, Xiangliang Zhang, Haitao Mi, and Dong Yu. Too correct to learn: Reinforcement learning on saturated reasoning data. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 205–215, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-short.19. URL https://aclanthology.org/2026.acl-short.19/.

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Rlvr without ineffective samples: Group prioritized off-policy optimization for llm reasoning, 2026. URL https://arxiv.org/abs/2606.01281.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KJvHnl3kUv.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *The Thirty-ninth Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yfYgwjj5F8.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. 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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. SimpleRL-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vSMCBUgrQj.

Zhi Zhang, Zhen Han, Costas Mavromatis, Qi Zhu, Yunyi Zhang, Sheng Guan, Dingmin Wang, Xiong Zhou, Shuai Wang, Soji Adeshina, Vassilis Ioannidis, and Huzefa Rangwala. Train less, learn more: Adaptive efficient rollout optimization for group-based reinforcement learning, 2026. URL https://arxiv.org/abs/2602.14338.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=neZSGqhxDa.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A LIMITATIONS

Our study has several limitations. First, the experiments mainly focus on mathematical and reasoning benchmarks with verifiable rewards. It remains unclear whether the same saturation-aware interventions transfer to open-ended tasks where correctness is harder to verify. Second, our systematic experiments use Qwen3 and LLaMA3.1, while broader scaling studies across larger model families are further needed. Third, wrong-answer rollouts depend on the quality and diversity of the generated incorrect trajectories. Poorly constructed negatives may introduce noise or encourage superficial contrast rather than robust reasoning. Finally, our iterative saturation mining results are preliminary and run for a limited number of steps. Future work should study longer training runs, stronger format control, and more adaptive criteria for deciding when newly saturated prompts should be added back into training.

B EXPERIMENTAL DETAILS

B.1 SYSTEM PROMPT

For all experiments, we used the following system prompt to guide the model's generation format, ensuring that it produces a step-by-step reasoning process and a clearly marked final answer (Zeng et al., 2025):

System Prompt

```
Please reason step by step, and put your final answer within
\boxed{}.
```

B.2 WRONG ANSWER PROMPT

Wrong Answer Suffix

```
Now solve the problem as a student with one specific,
subtle, and plausible mathematical misconception.  Privately
choose the misconception first, but do not explicitly mention
it.  Write a coherent step-by-step solution that consistently
follows this misconception, keep the final answer in \boxed{},
and make the final answer incorrect.
```

B.3 HYPERPARAMETER SETTINGS

We utilize GRPO for post-training. The model is optimized using AdamW with a learning rate of 1×10^{-6} and a weight decay of 1×10^{-2}. The training batch size is set to 128, and the maximum response length is 4096. For GRPO-specific configurations, we use a group size of $G = 8$ and enable KL regularization with coefficient $\beta = 1 \times 10^{-3}$. Unless otherwise specified, rollouts are sampled with a temperature of 1.0, and models are trained for 2 epochs.

For intervention-specific configurations, the higher-temperature variant uses a sampling temperature of $t = 1.5$, while negative rollout construction uses $G_{\text{wrong}} = 2$ constructed negative trajectories per saturated group. For reward-level interventions, we use `Skywork/Skywork-Reward-V2-Qwen3-8B` as the reward model and `Qwen/Qwen3-8B` as the LLM judge. For iterative saturation mining, every 10 training steps, we resample $G = 8$ rollouts from the current policy on the remaining training prompts and add prompts whose rollouts all receive correct rewards to the saturated pool. Table 5 summarizes the main hyperparameters.

Table 5: Key hyperparameters used.

Category	Value
Group Size (G)	8
Training Batch Size	128
PPO Mini-batch Size	32
KL Loss Type	Low-variance KL
KL Coefficient (β)	1×10^{-3}
Rollout Top-p	1.0
Rollout Temperature	1.0
Validation Top-p	0.95
Validation Temperature	1.0
Learning Rate	1×10^{-6}
Weight Decay	1×10^{-2}
Max Response Length	4096
Epochs	2
λ	0.5

Table 6: Robustness across three random seeds on Qwen3-1.7B. We report individual runs together with the mean and standard deviation across seeds.

Settings	MATH500	GPQA	BBH	Minerva	AIME24	AIME25	IFBench	IFEval	Avg.
GRPO									
Seed 1	62.60	24.24	10.86	17.28	17.08	9.17	21.20	69.00	28.93
Seed 2	64.00	22.22	12.72	16.18	19.90	10.73	22.00	68.00	29.47
Seed 3	63.60	23.23	12.43	17.28	19.38	8.02	24.40	69.60	29.74
Mean ± Std.	63.40 ± 0.72	23.23 ± 1.01	12.00 ± 1.00	16.91 ± 0.64	18.79 ± 1.50	9.31 ± 1.36	22.53 ± 1.67	68.87 ± 0.81	29.38 ± 0.41
Negative Rollout									
Seed 1	64.80	29.80	13.01	18.38	20.31	14.38	24.00	67.60	31.54
Seed 2	65.20	31.31	12.52	18.01	21.68	14.90	24.80	66.60	31.88
Seed 3	65.80	30.81	13.05	19.85	23.96	13.96	26.00	68.20	32.70
Mean ± Std.	65.27 ± 0.50	30.64 ± 0.77	12.86 ± 0.30	18.75 ± 0.97	21.98 ± 1.84	14.41 ± 0.47	24.93 ± 1.01	67.47 ± 0.81	32.04 ± 0.60

C Robustness Across Random Seeds

We further evaluate the robustness of Negative Rollout to training randomness. Specifically, we train Qwen3-1.7B using both standard GRPO and Negative Rollout with three random seeds and report the results in Table 6. Negative Rollout consistently outperforms GRPO across all three runs, improving the average score from 29.38 ± 0.41 to 32.04 ± 0.60. This consistent improvement across independent runs suggests that the gains from Negative Rollout are robust to training randomness rather than being specific to a single training seed.

D Further Analysis of Supervised Fine-Tuning

As shown in Table 1, SFT and SFT→GRPO do not consistently outperform standard GRPO, whereas Negative Rollout achieves stronger aggregate performance. To better understand this difference, we examine how SFT changes individual predictions. On MATH, SFT corrects 25 previously incorrect examples but also breaks 25 previously correct ones. On GPQA, it fixes 51 examples while breaking 51, and on AIME24, it fixes 33 while breaking 67. Thus, although SFT substantially changes model behavior, these changes do not consistently translate into improved reasoning accuracy.

One possible explanation is that SFT provides only positive imitation targets. In saturated settings, the model already produces successful trajectories, so additional teacher trajectories provide limited information about which behaviors should be strengthened or avoided. In contrast, Negative Rollout introduces both successful and unsuccessful trajectories for the same prompt, providing trajectory-level distinctions that are absent from positive-only imitation.

E A LARGER ADVANTAGE DOESN'T LEAD TO BETTER PERFORMANCE

E.1 FULL DERIVATION

We derive the effect of adding zero-reward samples to a saturated GRPO group. Consider a group where all original rollout rewards are correct:

$$\underbrace{1, 1, \ldots, 1}_{G \text{ samples}}. \tag{5}$$

The group has zero variance, so standard group normalization produces no useful relative advantage. We append k artificial zero-reward samples:

$$\underbrace{1, 1, \ldots, 1}_{G \text{ samples}}, \underbrace{0, 0, \ldots, 0}_{k \text{ samples}}. \tag{6}$$

Let $n = G + k$. Ignoring the small numerical ϵ in the denominator, the group mean is

$$\mu = \frac{G}{G+k}. \tag{7}$$

Using the sample standard deviation with denominator $n - 1$, the variance is

$$\sigma^2 = \frac{1}{n-1}\left[G(1-\mu)^2 + k(0-\mu)^2\right] \tag{8}$$

$$= \frac{1}{n-1}\left[G\left(\frac{k}{n}\right)^2 + k\left(\frac{G}{n}\right)^2\right] \tag{9}$$

$$= \frac{Gk}{(G+k)(G+k-1)}. \tag{10}$$

Thus,

$$\sigma = \sqrt{\frac{Gk}{(G+k)(G+k-1)}}. \tag{11}$$

For an original reward-1 sample, the normalized advantage is

$$A_1(k) = \frac{1-\mu}{\sigma} = \frac{\frac{k}{G+k}}{\sqrt{\frac{Gk}{(G+k)(G+k-1)}}} = \sqrt{\frac{k(G+k-1)}{G(G+k)}}. \tag{12}$$

For an added zero-reward sample, the normalized advantage is

$$A_0(k) = -\sqrt{\frac{G(G+k-1)}{k(G+k)}}. \tag{13}$$

The per-correct-sample advantage $A_1(k)$ is monotonically increasing in k. To see this, maximize its square and ignore the constant $1/G$:

$$f(k) = \frac{k(G+k-1)}{G+k} = k - \frac{k}{G+k}. \tag{14}$$

Then

$$f'(k) = 1 - \frac{G}{(G+k)^2} > 0 \tag{15}$$

for $G > 1$ and $k \geq 0$. Therefore, maximizing per-sample advantage alone would suggest adding arbitrarily many zero-reward samples.