ADVANTAGE HARNESS: REGULATING GRPO'S ADVANTAGE SIGNAL BY SHAPING WHAT IT SEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) is widely studied for reinforcement learning with verifiable rewards, where its advantage estimator assigns each rollout a magnitude from within-group reward statistics. This magnitude does not tell a reasoned correct answer from a guessed one, and it is largest when few rollouts in the group are correct. We identify this component as the *spurious advantage*. It arises on bounded-answer tasks and on the bounded sub-cases that cover more than half of MATH-7.5K. On multiple-choice prompts, about half of the correct rollouts in single-correct groups lack a supporting derivation, and training reduces this share but does not remove it. A zero-mean reweighting that gives one value to correct and one to wrong rollouts only rescales the group and cannot act on groups whose rollouts are all correct or all wrong. The lever that remains is the prompt distribution, which GRPO keeps fixed while the pass rate of the policy, and with it the set of groups that carry a gradient, moves during training. To this end, we propose Advantage Harness, a prompt selection layer that over-draws prompts at each update and keeps those least likely to yield a zero-gradient group. The estimator and the training rollout budget remain those of GRPO. On Qwen2.5-0.5B, the harness halves the zero-gradient rate and improves the eight-benchmark average over GRPO, a shuffled control and a difficulty-label filter on every seed. On Qwen2.5-3B, a prior rebuilt from a checkpoint after 40 updates restores the zero-gradient reduction. Code will be released.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the standard recipe for post-training large language models on reasoning tasks, and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is widely studied in this line, where a group of rollouts is sampled per prompt, each rollout is scored by a binary verifier, and the per-rollout advantage is constructed from within-group reward statistics (Shao et al., 2024; Guo et al., 2025). Recent improvements operate around this estimator, *e.g.*, refining the clip range, replacing or removing the normalization term, reshaping the sampling strategy, *etc.* (Yu et al., 2026; Liu et al., 2025; Xiao et al., 2025; Wu et al., 2025a).

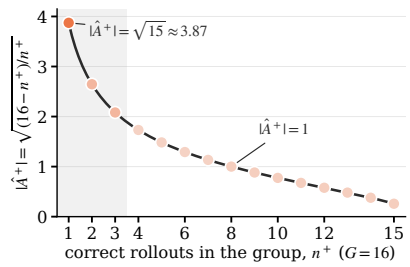

Figure 1: GRPO's magnitude $|\hat{A}^+|$ against the correct count n^+ with the group number G=16. Shaded area means $n^+ \leq 3$.

The GRPO advantage estimator applies the same within-group normalization formula on every task, where the magnitude assigned to each rollout is determined by the in-group correct/wrong composition, regardless of whether the prompt is an open-answer math problem or a multi-choice question. In the common case, a correct rollout earns its magnitude because the policy reasoned through the problem, which serves as an effective weight for the gradient. *Yet, an overlooked case shares the same surface*: a rollout can land on the correct answer by guessing one candidate answer rather than by reasoning, and still receives the same high magnitude. The estimator does not distinguish the two. The per-rollout gradient weight then contains a component that comes from guessing, misleading the policy toward learning lucky-guess trajectories instead of reasoning. We call this component the *spurious advantage* (Figure 1).

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.

Interestingly, this phenomenon is not exclusive to bounded-answer benchmarks; it extends to other settings whose effective answer surface remains finite. On the training-data side, even an open-answer math corpus can be far from uniformly open: in MATH-7.5K, 55.95% of problems admit short numeric or categorical answers, so the corpus is in fact a mixture of many small bounded subcases (Table 10), leaving the policy a non-negligible probability of guessing into the gold answer. This component is measurable. On multiple-choice prompts, we check whether the derivation of each correct rollout supports its answer: before training, about half of the correct rollouts in groups with a single correct rollout lack a supporting derivation, and these are the rollouts to which the estimator assigns its largest magnitude. Training reduces this share but does not remove it, and the hardest prompts keep the highest share.

A straightforward way to remove it is to change the weights inside the estimator. However, a zero-mean reweighting that gives one value to correct and one value to wrong rollouts must balance the two sides of every group, so it keeps their relative weights and only rescales the whole group. It cannot act on groups whose rollouts are all correct or all wrong, about a third of the groups GRPO draws at 0.5B, nor change which groups are drawn. Per-group rescaling, tested with Dr. GRPO (Liu et al., 2025), RLOO (Ahmadian et al., 2024) and our reweighting variant, remains within the seed noise of GRPO. Which groups the estimator receives is the choice that remains.

To this end, we propose Advantage Harness, a prompt selection layer between the training data and the estimator. The layer treats the prompt distribution as an input of the optimization that follows the pass statistics of the policy, rather than a constant fixed before training. At each update it over-draws candidate prompts, keeps those least likely, by one offline pass of the policy, to be all wrong or all correct, and generates rollouts only for them. Unlike dynamic sampling in DAPO (Yu et al., 2026), the harness selects before generation, so the estimator, the PPO-style surrogate and the training rollout budget remain those of GRPO. A shuffled control permutes the same prior values across prompts and separates the information in the prior from the effect of over-drawing. Since a zero-gradient group also arises when a prompt becomes too easy during training, we further study a prior rebuilt from a checkpoint after 40 updates and used from the first update of a fresh run. The harness acts on which groups reach the estimator, not on how the estimator reads a rollout, and Section 4.3 measures the two effects separately. We summarize the contributions as follows:

- This work isolates a *spurious advantage* phenomenon in the GRPO advantage estimator, which can mislead the policy toward guess-like behaviors rather than reasoning, and quantifies it on a mainstream math corpus and on real rollouts.
- This work shows that a zero-mean two-valued reweighting of the estimator only rescales a group and cannot act on this component, which explains why GRPO, Dr. GRPO, RLOO and our reweighting variant stay within seed noise of one another.
- This work proposes Advantage Harness, a prompt selection layer that keeps the estimator and the PPO-style surrogate unchanged, samples no additional rollouts during training and pays for its prior with one offline pass per model, with an analysis that links the selection to the advantage mass of a group and to the noise of the update.
- Experiments on math reasoning at two model scales, with a shuffled control, a difficulty-label control and multiple seeds, show that the harness reduces the zero-gradient rate at both scales and improves the benchmark average at 0.5B on every seed. We hope this work may invite further attention to how GRPO interacts with the structure of specific tasks.

2 Related Work

GRPO and Its Variants. GRPO (Shao et al., 2024) estimates per-rollout advantage from the in-group reward distribution and removes the PPO value network, and has driven strong results on open-answer math reasoning at scale (Shao et al., 2024; Guo et al., 2025). Follow-up work has refined the surrounding mechanics on clipping and sampling (Yu et al., 2026; Xiong et al., 2025), on normalization and aggregation (Liu et al., 2025; Xiao et al., 2025; Wu et al., 2025a; Zhao et al., 2025; He et al., 2025c; Pang et al., 2025), on advantage shaping (Wang et al., 2025; Huang et al., 2025; Chen et al., 2025; Zhang et al., 2025), on the treatment of wrong rollouts and all-wrong groups (Zhu et al., 2026; Nan et al., 2025; Feng et al., 2025; Li et al., 2025a), and on multi-objective composite rewards (Ichihara et al., 2025). Their evaluations center on open-answer math, where the GRPO magnitude can be treated as a valid credit signal on every prompt. GRPO, Dr.GRPO, RLOO and

BNPO assign the magnitude from the reward counts of a group; together with our reweighting variant they form the two-valued zero-mean family of Section 3.4, whereas REINFORCE++ normalizes over the whole batch and lies outside it.

Advantage Estimators. A complementary line of work analyzes the GRPO advantage estimator. Mroueh (2025) derives the closed-form weighting that GRPO implicitly applies under binary rewards, $\omega^+(p) = \sqrt{(1-p)/p}$ and $\omega^-(p) = -\sqrt{p/(1-p)}$, and shows that GRPO is equivalent to an adaptive weighted contrastive loss in which rare-success rollouts receive large positive weight; this adaptive weighting is a *feature* on open-answer math, where rare success corresponds to a rare reasoning trajectory. Related analyses characterize the estimator from cross-prompt expectation (Yang et al., 2026), rank-bias (He et al., 2025a), KL-constrained closed-form (Zhang et al., 2026), group-size scaling (Wu et al., 2025b), supervised-learning reframings (Li et al., 2025b; 2026) angles. We adopt Mroueh's closed-form as the starting point of our analysis. While the works above develop theory of the estimator within the regime where the within-group composition reflects reasoning, our analysis focuses on the complementary regime: we identify two structural cases in which this composition acquires a reasoning-independent component, measure the component on real rollouts, and show that no reweighting of the estimator can remove it, which motivates selecting the groups the estimator receives.

Prompt and rollout selection. A second direction changes the prompts the estimator receives. DAPO (Yu et al., 2026) generates the groups first, removes those whose accuracy is 0 or 1, and draws further prompts until the batch reaches the target size, which raises the rollout count (Section 4). Reinforce-Ada (Xiong et al., 2025) and GVM-RAFT (Yao et al., 2026) assign more rollouts during generation to prompts with uncertain pass rate. Prompt curriculum learning (Gao et al., 2026) selects before generation with a value model updated alongside the policy. GRESO (Zheng et al., 2026) skips a prompt before generation when its reward history from earlier epochs predicts a zero-variance group. Skywork-OR1 (He et al., 2025b) removes prompts the initial model always or never solves, and repeats this per stage. The harness is a pass-rate filter of this kind applied per update: it selects before generation without additional rollouts during training, at the one-time cost of an offline pass, which Section 4 counts separately. Beyond the filter itself, this work adds a shuffled control, a measurement of the magnitude distribution the estimator receives, a stale-versus-refreshed comparison of the prior, and an analysis of how the selection enters the advantage and the update.

Correctness without reasoning. An outcome reward marks any rollout whose final answer matches the reference as correct. Uesato et al. (2022) report that outcome-based supervision matches process-based supervision on final-answer error. Reducing the share of correct answers with incorrect reasoning requires process-based feedback (Uesato et al., 2022; Lightman et al., 2024). Balepur et al. (2024) show that models answer multiple-choice questions above chance from the choices alone. Shao et al. (2025) show that Qwen2.5-Math gains on MATH-500 under random rewards. Reward hacking (Ichihara et al., 2025; Fu et al., 2025) exploits a composite reward; in both cases the reward is the cause, whereas here the reward is correct and the derivation is not. We treat a correct rollout without a derivation as a property of the prompt and select prompts before they reach the estimator.

3 METHOD

3.1 PRELIMINARY

Given a question q and a current policy π_θ, GRPO (Shao et al., 2024) samples a group of G rollouts $\{\tau_1, \ldots, \tau_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and assigns each rollout a binary verifier reward $r_i \in \{0, 1\}$. We write n^+ and $n^- = G - n^+$ for the in-group counts of correct and wrong rollouts. The per-rollout advantage is computed by within-group standardization,

$$\hat{A}_i = \frac{r_i - \mu}{\sigma + \varepsilon}, \tag{1}$$

where $\mu = \frac{1}{G}\sum_j r_j$, $\sigma^2 = \frac{1}{G}\sum_j (r_j - \mu)^2$, and ε is a numerical stabilizer. The policy is then updated with the PPO-style clipped surrogate (Schulman et al., 2017), in which $|\hat{A}_i|$ is the per-rollout scalar weight on rollout i's gradient contribution to the policy update (see Appendix B).

With $r_i \in \{0, 1\}$, the group statistics reduce to $\mu = n^+/G$ and $\sigma = \sqrt{n^+ n^-}/G$, and substituting into Eq. (1) gives

$$\hat{A}^+ = \sqrt{n^-/n^+}, \quad \hat{A}^- = -\sqrt{n^+/n^-}, \tag{2}$$

which has also been derived in concurrent work (Mroueh, 2025). Thus, the magnitude depends only on (n^+, n^-), not on the rollout content (Figure 1). A group with $n^+ \in \{0, G\}$ has $\sigma = 0$ and $\hat{A}_i = 0$, which we denote as a *zero-gradient group*.

3.2 Impact of the GRPO advantage estimator

The magnitude in Eq. (2) is largest in imbalanced groups. For $G = 16$, a rare-correct group ($n^+ = 1$) gives $|\hat{A}^+| = \sqrt{15} \approx 3.87$, while a balanced group ($n^+ = 8$) gives $|\hat{A}^+| = 1$. The rare-correct rollout thus receives 3.87 times the weight of a rollout in a balanced group.

To examine what learning signal this amplification carries, we decompose the per-prompt success probability p_q by the source of the $+1$ outcome:

$$\underbrace{p_q}_{\text{prob. of } r=+1} = \underbrace{p_{\mathrm{r}}}_{\text{from reasoning}} + \underbrace{(1 - p_{\mathrm{r}}) \cdot p_{\mathrm{g}}}_{\text{from non-reasoning}}, \tag{3}$$

where p_{r} is the probability that the policy obtains the correct answer through reasoning, and p_{g} is the probability of obtaining the correct answer in any other way (e.g., by naming one of a few candidate answers). The count $n^+ \sim \mathrm{Binomial}(G, p_q)$ inherits both terms, so the magnitude $\sqrt{n^-/n^+}$ carries a component driven by reasoning and one driven by the second term of Eq. (3). We call the latter the *spurious advantage*.

When $p_{\mathrm{g}} \approx 0$, the second term of Eq. (3) vanishes and the spurious advantage is small. This is the case for open-answer mathematical reasoning (Mroueh, 2025; Shao et al., 2024; Guo et al., 2025) (Appendix A); the next subsection works out two cases where p_{g} is not small.

3.3 Cases with a large spurious advantage

Bounded answer sets. When the task admits only a small set of candidate answers (e.g., multi-choice with k options), a policy that emits one candidate at random hits the correct answer with probability $p_{\mathrm{g}} \approx 1/k$. For 4-choice prompts with $G = 16$, guessing alone yields $G \cdot p_{\mathrm{g}} = 4$ correct rollouts in expectation, yet a group with $n^+ = 1$ receives the largest magnitude $\sqrt{15}$. By Eq. (3), the expected fraction of this magnitude that comes from reasoning is $p_{\mathrm{r}}/(p_{\mathrm{r}} + (1 - p_{\mathrm{r}})/k)$, which is small whenever p_{r} is comparable to or below $1/k$ (Appendix A).

An open-answer training set can also exhibit the bounded case on a large subset of its problems: grouping the gold answers of MATH-7.5K by syntactic shape, five of 11 categories are bounded and cover 55.95% of the problems, with the highest share on the easiest ones (Appendix D, Table 10).

Measuring the non-reasoning share on real rollouts. To measure the second term of Eq. (3) on real rollouts, we use a *derivation check* for multiple-choice prompts. Specifically, for a rollout marked correct, the check tests whether the last number written before the final answer statement equals the value of the chosen letter; a correct rollout that fails this test, or that contains no number, is labeled flawed (Appendix F gives the rule and its validation against an LLM judge).

We apply the check to 8,000 AQuA-RAT training prompts (Ling et al., 2017) with $G = 16$ rollouts each, before training and after 200 GRPO updates of one run (seed 1) trained on the same prompts. Table 1 fixes the prompt sets before training, so that bucket migration does not confound the comparison (unmatched buckets in Appendix F). Prompts with a single correct rollout before training reach a mean k of 5.0 after 200 updates, and the flawed share of their correct rollouts decreases from 52.2% to 39.5%. Prompts with $k \geq 8$ before training decrease from 40.7% to 29.3%. The estimator meets these rollouts with its largest magnitude: on the untrained policy, flawed rollouts are 40.9% of the correct rollouts, but they carry 57.9% of the positive advantage weight once each correct rollout is weighted by the $|\hat{A}^+|$ of its group (Table 5 in Appendix F).

Prompt set	prompts	before training		after 200 updates		mean k
		correct	flawed↓	correct	flawed↓	after
k=1 before	1432	1432	52.2%	7157	39.5%	5.0
$k\geq$8 before	734	6945	40.7%	7614	29.3%	10.4

Table 1: Prompt-matched derivation check on the same 8,000 AQuA-RAT prompts (the 0.5B model, G=16). Prompt sets are fixed before training by their number k of correct rollouts; columns give the number of prompts, the number and flawed share of their correct rollouts before and after 200 GRPO updates, and the mean k after training.

3.4 Limits of within-group reweighting

A first response to Section 3.2 is to change the magnitude inside the estimator. Consider any estimator that assigns $a^+(n^+) > 0$ to every correct rollout of a group and $a^-(n^+) < 0$ to every wrong one. With zero group mean, we have

$$\tilde{A}_i = \begin{cases} a^+(n^+), & r_i = 1, \\ a^-(n^+), & r_i = 0, \end{cases} \qquad \sum_{i=1}^{G} \tilde{A}_i = 0. \tag{4}$$

Zero mean forces

$$n^+ \, a^+ \;=\; n^- \, |a^-|, \tag{5}$$

so $a^-/a^+ = -n^+/n^-$ regardless of the choice of a^+. Thus, the correct and wrong rollouts of a group keep their relative weights under GRPO. The group as a whole is rescaled by $\lambda(n^+) = a^+(n^+)\sqrt{n^+/n^-}$ relative to GRPO.

This has four consequences for the spurious advantage. (1) A correct rollout without a derivation and a correct rollout with one receive the same weight under every member of the family, since the family reads the reward alone. (2) A constant positive magnitude does not remove the count dependence; Eq. (5) moves it to the wrong side. (3) A zero-gradient group keeps a zero advantage under any such reweighting, and such groups are 36.1% of GRPO's groups at 0.5B and 48.0% at 3B. (4) The reweighting cannot change which groups are drawn, so the only free quantity is the per-group scalar $\lambda(n^+)$, i.e., a count-dependent learning rate. GRPO, Dr. GRPO (Liu et al., 2025) and RLOO (Ahmadian et al., 2024) are members of this family; so is the reweighting variant we constructed for this problem ($a^+ = c$). Empirically, all three remain within the seed noise of GRPO (Section 4.5). Thus, reweighting changes at most the per-group scale, which motivates Section 3.5.

3.5 Proposed method: Advantage Harness

The estimator of Eq. (1) gives a non-zero advantage only to groups with $0 < n^+ < G$, so an update reads a composition distribution set by the prompt distribution and the pass rate of the current policy, which moves as the policy learns. We therefore treat the prompt distribution as an input of the optimization and insert a layer between the data and the estimator, leaving the estimator, the surrogate and the training rollout budget of GRPO unchanged. At update t, it draws Mb candidates $\mathcal{C}_t$ ($M = 3$), scores them by a lookup prior, keeps the b with the smallest score and generates G rollouts only for them (Figure 2; Algorithm 2),

$$s(q) = \hat{P}\big(n^+ \in \{0, G\} \mid q\big), \qquad \mathcal{S}_t = \operatorname*{arg\,min}_{\mathcal{S} \subset \mathcal{C}_t,\ |\mathcal{S}|=b} \sum_{q \in \mathcal{S}} s(q). \tag{6}$$

From selection to the update. Selection reads only the prompt, so it changes how much advantage a drawn group carries, not that of a rollout. Under Eq. (2), the advantages of a group sum to zero and their total magnitude $\sum_i |\hat{A}_i| = 2\sqrt{n^+ n^-}$ vanishes if and only if $n^+ \in \{0, G\}$. With p_q the success probability on q, $n^+ \sim \mathrm{Binomial}(G, p_q)$, the zero-gradient probability is $z(q) = p_q^G + (1-p_q)^G$, and $\mathbb{E}[\sqrt{n^+ n^-} \mid q] = (1 - z(q))\, c(q)$ with $c(q) \in [\sqrt{G-1},\, G/2]$ (Appendix C). The score $s(q)$ of Eq. (6) estimates $z(q)$, the factor selection acts on. Zero-gradient groups are dropped before the loss and the surrogate averages over the kept rollouts (Appendix E), so one update moves θ along

$$\hat{g}_t \;=\; \frac{1}{N_t} \sum_{q \in \mathcal{S}_t} \mathbf{1}[0 < n_q^+ < G] \sum_{i=1}^{G} \hat{A}_{q,i}\, \bar{\nabla}_\theta \log \pi_\theta(\tau_{q,i}), \tag{7}$$

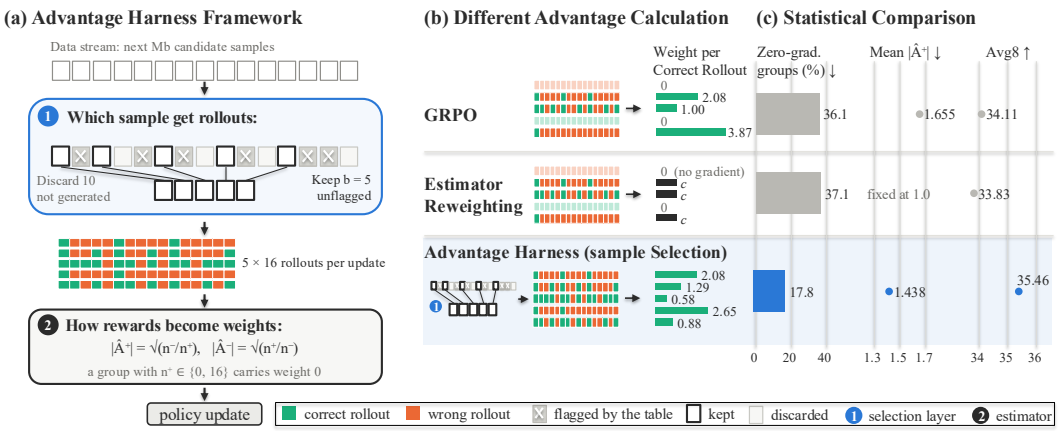

Figure 2: Overview of the proposed method. **(a)** One update under the harness: from the next Mb=15 candidates, a lookup table built by one offline pass of the initial policy (flag = all 16 rollouts wrong or all correct) marks the flagged samples; the b=5 unflagged ones with the lowest score are kept, the other 10 are discarded before any rollout is generated, and the update trains on 5×16 rollouts as in GRPO. **(b)** GRPO, estimator reweighting, and the proposed harness comparison (❶ selection layer, ❷ estimator). **(c)** Zero-gradient rate, mean $|\hat{A}^{+}|$ over trainable groups and Avg8.

where $\bar{\nabla}_\theta \log \pi_\theta(\tau)$ is the token-mean score of a rollout (Appendix B) and N_t the number of kept rollouts. Given q, a group contributes $(1 - z(q))\,\bar{g}(q)$, with $\bar{g}(q)$ its mean over trainable groups.

Proposition 1 (Harness as GRPO under a shifted prompt distribution; proof and caveats in Appendix C)**.** *Since the selection depends on q only, $z(q)$ and $\bar{g}(q)$ coincide with those of GRPO, and the harness is GRPO run on the prompt marginal $\mathcal{D}_H$ of a kept candidate, which does not depend on θ. Under Eq. (7), the update averages N_t^{grp} trainable groups with $\mathbb{E}[N_t^{\mathrm{grp}}] = b(1-\bar{z})$ and conditional variance $\sigma_g^2/N_t^{\mathrm{grp}}$, so at equal rollouts the expected number of trainable groups per update, and to first order the signal-to-noise ratio of the update, change by $(1-\bar{z}_H)/(1-\bar{z}_{\mathrm{GRPO}})$, where $\bar{z}$ is the zero-gradient rate of the drawn groups.*

Instantiations. The instantiations differ in the policy whose pass statistics $s(q)$ reads and how often it is re-estimated: (1) the offline prior, our main setting, from one pass of the initial policy; (2) the refreshed prior, built from a policy after 40 updates and used from the first update of a fresh run (Section 4.4); the online rule we propose, a rebuild once all-correct groups outnumber all-wrong ones among the drawn groups, is not run in this paper; (3) the online blend, $\frac{1}{2}\hat{P}(q) + \frac{1}{2}d_{\ell(q)}$ with d_ℓ an exponential moving average of the zero-gradient indicator of level ℓ under the current policy, re-estimated at every update as a curriculum (Section 4.5).

Prior construction. The prior $\hat{P}$ is built from one offline pass of the initial policy over the training prompts with G rollouts each. It records $\hat{P}(q) = 1$ if the group is all-wrong or all-correct and 0 otherwise. The score is a pass rate under the initial policy and says nothing about derivations.

Shuffled control. The shuffled control permutes the prior values at random across prompts and runs the same rule with the same M, which separates the effect of over-drawing alone.

4 Experiments

4.1 Settings

Models, data and compute. We train Qwen2.5-0.5B-Instruct and Qwen2.5-3B-Instruct (Qwen et al., 2025) on the 7,496 training prompts of MATH-7.5K (Hendrycks et al., 2021) with a rule-based binary verifier. Each run makes 200 updates with $b = 5$ prompts per update and $G = 16$ rollouts per prompt, i.e., 16,000 rollouts during

Benchmark	Mean	SD	2×SEM
GSM8K	50.57	0.45	0.53
MATH-500	55.20	1.22	1.40
Minerva	7.48	0.93	1.07
Olympiad	7.54	0.71	0.82
MMLU-Math	51.20	3.86	4.45
SAT-Math	61.07	2.39	2.76
AQuA	30.45	3.38	3.91
AMC23	9.36	1.51	1.74
Avg8	34.11	1.14	1.32

Table 2: Seed noise of GRPO on 0.5B model with results over three seeds.

Method	Open-answer				Bounded-answer				Avg8↑
	GSM8K	MATH-500	Minerva	Olympiad	MMLU-M	SAT-M	AQuA	AMC	
Untrained	47.23	48.40	8.46	10.25	52.54	55.27	32.23	7.30	32.71
GRPO	50.57±0.45	55.20±1.22	7.48±0.93	7.54±0.71	51.20±3.86	61.07±2.39	30.45±3.38	9.36±1.51	34.11±1.14
Dr. GRPO	49.89±0.66	56.93±1.96	7.72±1.33	7.89±0.71	52.09±3.14	60.09±2.16	29.94±3.20	9.44±1.51	34.25±1.00
RLOO	49.71±0.18	55.80±1.60	8.21±0.42	8.66±0.31	54.01±1.52	60.61±1.88	32.52±3.39	9.66±0.45	34.90±0.97
DAPO	50.34±1.07	55.80±1.44	6.99±0.97	7.95±0.98	52.81±0.80	60.35±1.95	29.69±0.82	8.56±0.81	34.06±0.38
Ours	49.68±0.53	55.40±0.69	8.95±0.21	8.83±0.64	54.10±0.76	62.11±1.01	35.33±2.30	9.24±0.81	35.46±0.35

Table 3: Accuracy (%, ↑) on the eight core benchmarks and Avg8 for the 0.5B model. RL rows report mean±SD over three seeds, and the untrained row one evaluation. DAPO runs its dynamic sampling under the same 16,000 training rollouts. We shade the proposed method.

training, a small per-update batch in which a zero-gradient group is a large share of an update. The harness samples no additional rollouts during training; its lookup prior costs one offline pass of 16 rollouts per prompt (119,936 rollouts at 0.5B), built once per model and shared by every run that reads it, i.e., 29,326 rollouts per run when amortized over the nine runs of this paper (Table 15). Hyperparameters are given in Appendix E. DAPO (Yu et al., 2026) runs its dynamic sampling until it reaches the same 16,000-rollout budget, so that every compared method spends the same number of training rollouts (Appendix E).

Seeds and reporting. On the 0.5B model, every setting has three seeds. On 3B, GRPO and the stale-prior harness have three seeds, its shuffled control two, the refreshed pair six each, the online blend two and DAPO one (Table 13). All numbers are final checkpoints; no checkpoint is selected on the test sets. Avg8 is the mean accuracy over the open-answer benchmarks GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), Minerva-Math (Lewkowycz et al., 2022) and OlympiadBench (He et al., 2024) and the bounded-answer benchmarks MMLU-Math (Hendrycks et al., 2020), SAT-Math (Zhong et al., 2024), AQuA (Ling et al., 2017) and AMC (Jia et al., 2024); the first five use greedy decoding and the last three the mean over 16 samples at temperature 0.6. Means carry the standard deviation over seeds, and paired differences are computed on the same seed (see seed noise analysis in Table 2).

4.2 Main Results on 0.5B

Table 3 compares the harness with GRPO, its two-valued variants and DAPO at 0.5B. The harness and the shuffled control share the seed, the over-draw factor and the multiset of prior values, so their paired difference isolates the information in the prior. The harness outperforms the control on each of the three seeds ($+1.54$ on average, Table 11), and the control does not improve over GRPO (33.91 vs. 34.11).

Overall, the harness reaches 35.46 ± 0.35 Avg8 against 34.11 ± 1.14 for GRPO; the gain of $+1.35$ is positive on every seed but within one standard deviation (Table 4). Per benchmark, the largest gains fall on AQuA ($+4.9$), Minerva-Math ($+1.5$) and OlympiadBench ($+1.3$), all outside their $2\times$SEM, while SAT-Math ($+1.0$), MMLU-Math ($+2.9$) and MATH-500 ($+0.2$) stay within theirs (Table 2). However, the harness is inferior to GRPO on GSM8K (49.68 vs. 50.57, beyond its $2\times$SEM). Besides, DAPO, which needs no pass, reaches 34.06 ± 0.38 under the same training budget (Table 15 lists every budget with the pass).

4.3 Mechanism

As shown in Table 4, the harness reduces the zero-gradient rate from 36.1% under GRPO to 17.8%, while the shuffled control stays at 36.6%. It also lowers mean $|\hat{A}^+|$ from 1.655 to 1.438 and the $n^+ \leq 3$ share from 35.1% to 27.1%, whereas the control does not; since the control over-draws equally without the prior, the shift comes from the prior (Figure 3).

Let $H(k)$ denote the share of drawn groups with k correct rollouts and $f(k)$ the flawed rate of bucket k in Table 17 (before-training column, an AQuA proxy). Each correct rollout of such a group carries $|\hat{A}^+(k)|$, so the expected share of positive weight on correct rollouts without a derivation is

$$s = \frac{\sum_k H(k)\, k\, |\hat{A}^+(k)|\, f(k)}{\sum_k H(k)\, k\, |\hat{A}^+(k)|}. \tag{8}$$

Setting	Avg8↑	Ours−row	p	Zero-grad.↓	Mean $\lvert\hat{A}^+\rvert$ ↓	$n^+ \le 3$ ↓	Spurious share
GRPO	34.11±1.14	+1.35±0.93	0.129	36.1%±1.2%	1.655±0.006	35.1%±0.7%	42.3%±0.0%
Reweighting	33.83±0.39	+1.63±0.69	0.055	37.1%±0.3%	fixed	37.3%±1.0%	42.5%±0.1%
Shuffled control	33.91±0.75	+1.54±0.83	0.084	36.6%±1.9%	1.735±0.036	37.0%±1.6%	42.5%±0.1%
Level-label filter	34.66±0.17	+0.80±0.21	0.022	23.3%±2.1%	1.326±0.055	23.6%±1.9%	41.7%±0.1%
Online blend	34.80±0.73	+0.65±0.43	0.118	15.0%±1.9%	1.411±0.046	26.6%±1.7%	41.8%±0.1%
Ours	35.46±0.35	+0.57±0.33	0.087	17.8%±1.7%	1.438±0.051	27.1%±1.0%	41.8%±0.1%

Table 4: Controls and ablations on the 0.5B model, three seeds per row. Ours−row is the paired difference on the same seed with its two-sided paired t-test p. Zero-grad. is the share of drawn groups with $n^+ \in \{0, 16\}$, and mean $|\hat{A}^+|$ and the $n^+ \le 3$ share are taken over trainable groups (Reweighting fixes $|\hat{A}^+| = 1$). Spurious share follows Eq. (8) with $f(k)$ from Table 17. We shade the proposed method.

Setting	All↓	k=1 ↓	k≥8 ↓	$\lvert\hat{A}^+\rvert$-weighted↓	n
Untrained	40.9%	53.4%	36.3%	57.9%	1
GRPO	37.5%±4.2%	42.8%±2.0%	36.4%±7.5%	54.9%±4.9%	3
Shuffled	37.3%±1.9%	45.4%±5.3%	36.7%±2.3%	54.1%±3.4%	3
Ours	34.9%±2.5%	43.5%±0.9%	33.5%±3.8%	51.1%±2.8%	3

Table 5: Derivation check (flawed shares, %) on the untrained model and the trained policies of the 0.5B model. Each policy generates 16 rollouts per prompt on the same 1,500 AQuA prompts, and the shares are pooled, restricted to groups with k=1 or $k \ge 8$, and weighted by $|\hat{A}^+(k)|$. Cells report mean±SD over three seeds.

Setting	Spurious share↓	Mass rel. GRPO↓	n
GRPO	42.3%±0.0%	1.000±0.020	3
Level-label filter	41.7%±0.1%	1.201±0.036	3
Shuffled control	42.5%±0.1%	0.987±0.025	3
Ours	41.8%±0.1%	1.313±0.029	3

Table 6: Expected spurious share of the positive weight, Eq. (8), and the spurious mass per drawn group relative to GRPO on the 0.5B model. Cells report mean±SD over n seeds, and Table 18 adds the 3B model.

Spurious share of the positive weight. On 0.5B this share is 42.3% under GRPO, 41.8% under the harness, 42.5% under the shuffled control and 41.7% under the level-label filter, and on 3B every setting lies between 41.2% and 41.6% (Table 6). The harness shifts the n^+ distribution (Table 14) but leaves this share the same, since $f(k)$ is nearly flat for $k \ge 2$. It raises the positive weight per drawn group to 1.31 times that of GRPO (Table 6). The total positive mass per drawn group rises by 1.33, i.e., 1.29 from the share of trainable groups times 1.03 from the mean magnitude of a trainable group (Table 9 in Appendix C), a count effect as Proposition 1 states. On the final policies (Table 5), the flawed share among correct rollouts is 37.5%, 34.9% and 37.3% for GRPO, the harness and the control, against 40.9% before training, and the $|\hat{A}^+|$-weighted share is 51.1% vs. 54.9% and 54.1%, within the seed noise. Thus the 0.5B gain comes with more trainable groups per rollout, not with a smaller spurious fraction.

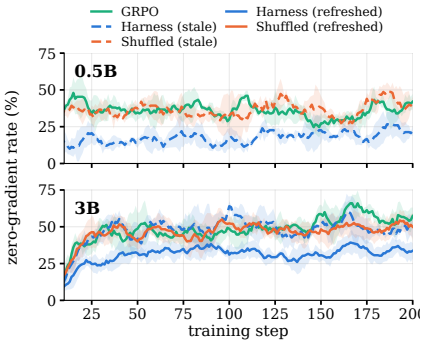

Figure 3: Zero-gradient rate comparison with 10-step moving average.

4.4 Refreshing the Prior on 3B

On 3B, the two quantities of Section 4.3 behave differently (Table 8). The refreshed prior reduces the zero-gradient rate from 48.0% to 30.9%, while its shuffled control reaches 46.0% (Table 14 in Appendix F). However, it keeps mean $|\hat{A}^+|$ close to that of GRPO (1.094 vs. 1.100) and gives no accuracy gain: GRPO reaches 54.43 ± 1.94, and the refreshed pair 53.09 ± 3.02 and 53.54 ± 3.76 (paired -0.44 ± 2.79, positive on three of six seeds, Table 13 in Appendix F). By comparison, the stale prior reduces mean $|\hat{A}^+|$ to 0.898 ± 0.035 with the zero-gradient rate close to that of GRPO (47.2% vs. 48.0%). It also has the highest Avg8 of Table 8, 56.41 ± 1.67 against 54.43 ± 1.94 for GRPO and 50.01 ± 1.96 for its shuffled control, within seed standard deviations of 1.67 to 6.08 across the 3B settings. The refresh thus restores the zero-gradient reduction at 3B (Corollary 1), while the magnitude shift and the accuracy gain appear at 0.5B only.

Waste composition and prior coverage. Figure 4 reports the all-wrong and all-correct shares among the zero-gradient groups drawn by GRPO and the share of these groups whose prompt each

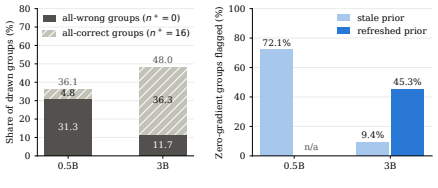

Figure 4: Waste composition and prior coverage. Left: all-wrong and all-correct shares among the groups GRPO draws. Right: share of these groups whose prompt each prior flags.

Estimator	Avg8 per seed	Avg8↑	Zero-grad. rate
GRPO	34.17/35.22/32.93	34.11±1.14	36.1%±1.2%
Dr. GRPO	33.75/35.40/33.59	34.25±1.00	37.5%±1.1%
RLOO	33.78/35.51/35.40	34.90±0.97	37.5%±1.6%
Reweighting	33.37/34.06/34.04	33.83±0.39	37.1%±0.3%

Table 7: Estimators of the two-valued zero-mean family (Section 3.4) on the 0.5B model under the same 16,000 training rollouts. We list Avg8 per seed, its mean±SD and the zero-gradient rate, which no reweighting can change.

Method	Open-answer				Bounded-answer				Avg8↑	Zero-grad. rate↓	Mean $\lvert\hat{A}^+\rvert$
	GSM8K	MATH-500	Minerva	Olympiad	MMLU-M	SAT-M	AQuA	AMC			
Untrained	73.77	52.40	16.54	26.86	21.12	30.27	24.24	27.26	34.06	–	–
GRPO	84.71±1.05	86.40±0.53	27.57±0.64	31.21±1.00	59.45±8.30	79.36±6.23	33.87±6.27	32.86±0.48	54.43±1.94	48.0%±0.8%	1.100±0.044
DAPO	86.05±1.04	88.10±0.77	26.76±1.06	29.21±1.00	66.10±10.67	79.00±9.56	27.30±8.24	35.24±1.33	54.86±2.35	49.4%±1.4%	1.069±0.035
Online blend	85.03±0.48	89.20±0.28	29.78±1.04	30.48±1.12	56.95±17.39	73.44±16.85	29.32±9.45	34.53±2.61	53.59±6.08	38.6%±2.1%	1.073±0.035
Shuffled control	84.53±0.43	86.50±0.42	27.94±1.56	30.04±0.25	47.73±9.26	63.57±10.63	25.50±5.76	34.22±0.27	50.01±1.96	48.4%±1.3%	1.154±0.038
Ours	83.83±0.94	87.53±1.17	27.21±1.91	30.04±0.98	65.64±6.85	81.25±4.88	42.22±7.25	33.61±1.33	56.41±1.67	47.2%±0.5%	0.898±0.035

Table 8: Accuracy (%, ↑) on the eight core benchmarks and Avg8 for the 3B model, with the zero-gradient rate and the mean $|\hat{A}^+|$ over trainable groups of the same runs. GRPO and Ours have three seeds, the online blend and the shuffled control two, and DAPO one, as listed per seed in Table 13. DAPO runs under the same training rollouts as the other methods. We shade the proposed method.

prior flags. The three GRPO seeds at 0.5B draw 36.1% zero-gradient groups, 31.3% all-wrong and 4.8% all-correct; at 3B the shares are 48.0%, 11.7% and 36.3%. The stale prior flags 36.5% of the prompts at 0.5B and covers 72.1% of the zero-gradient groups, but flags 6.3% and covers 9.4% at 3B; the refreshed prior flags 31.2% and covers 45.3% (Table 16 in Appendix F). At 3B, three of every four zero-gradient groups are all-correct groups, which appear only during training, and the refreshed prior covers a larger share of them than the stale prior.

4.5 Ablations

Table 4 collects the controls and ablations on 0.5B. On the estimator route, no member of the two-valued family changes the zero-gradient rate or improves Avg8 over GRPO (34.11 ± 1.14), since each changes only the scalar weight of a group (Table 7).

On the data route, DAPO's filter after generation keeps GRPO's magnitude distribution (35.8% zero-gradient groups at generation, mean $|\hat{A}^+|$ of 1.704), whereas selecting before generation changes it. The harness outperforms the level-label filter by $+0.80$ ($p = 0.022$), which measures what the offline pass adds beyond the difficulty label. The online blend lowers the zero-gradient rate further (15.0%) within the seed noise of the harness, whereas the shuffled control stays at GRPO's zero-gradient rate and below its Avg8. A calibration arm that reads the same prior inside the estimator is in Appendix F (Table 20).

5 Conclusion

We identified a *spurious advantage* component in the GRPO advantage estimator: whenever a task admits correct answers without reasoning, GRPO's count-dependent magnitude gives that component its largest weight in the groups where it is most frequent. A derivation check showed that training reduces, but does not remove, the correct rollouts without a derivation. Reweighting inside the estimator cannot change the relative magnitudes within a group. We therefore proposed Advantage Harness, which selects prompts before generation by a lookup prior at the training rollout budget of GRPO. On 0.5B the harness reduced the zero-gradient rate and improved Avg8 over GRPO and a shuffled control on every seed. On 3B the prior of the initial policy gave the highest Avg8, and the refreshed prior restored the reduction of zero-gradient groups. Proposition 1 explains the gain as a count effect, since at equal training rollouts the harness raises the expected number of trainable groups per update by 1.29 on the 0.5B model, which the measured advantage mass confirms. The harness changes how many groups carry a gradient, not the fraction of that gradient that is spurious.

AI USE STATEMENT

In this work, we used generative AI tools for designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, supporting qualitative and thematic data analysis, interpreting results, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, and providing critical ingredients for proving mathematical claims. Generating synthetic data sets and cleaning and reformatting dataset are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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. In *ACL*, 2024.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. Artifacts or abduction: How do llms answer multiple-choice questions without the question? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10308–10330, 2024.

Guanxu Chen, Yafu Li, Yuxian Jiang, Chen Qian, Qihan Ren, Jingyi Yang, Yu Cheng, Dongrui Liu, and Jing Shao. Conditional advantage estimation for reinforcement learning in large reasoning models. *arXiv preprint arXiv:2509.23962*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yunzhen Feng, Parag Jain, Anthony Hartshorn, Yaqi Duan, and Julia Kempe. Don't waste mistakes: Leveraging negative rl-groups via confidence reweighting. *arXiv preprint arXiv:2510.08696*, 2025.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. Reward shaping to mitigate reward hacking in rlhf. *arXiv preprint arXiv:2502.18770*, 2025.

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*, volume 2026, pp. 58614–58652, 2026.

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. In *arXiv preprint arXiv:2501.12948*, 2025.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *EMNLP*, 2025a.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *ACL*, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025b.

Zhiyuan He, Xufang Luo, Yike Zhang, Yuqing Yang, and Lili Qiu. δl normalization: Rethink loss aggregation in rlvr. *arXiv e-prints*, 2025c.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2026.

Wenke Huang, Quan Zhang, Yiyang Fang, Jian Liang, Xuankun Rong, Huanjin Yao, Guancheng Wan, Ke Liang, Wenwen He, Mingjun Li, et al. Mapo: Mixed advantage policy optimization. *arXiv preprint arXiv:2509.18849*, 2025.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. Mo-grpo: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *arXiv preprint arXiv:2509.22047*, 2025.

LI Jia, Beeching Edward, Tunstall Lewis, Lipkin Ben, Soletskyi Roman, Huang Shengyi Costa, Rasul Kashif, Yu Longhui, Jiang Albert, Shen Ziju, Qin Zihan, Dong Bin, Zhou Li, Fleureau Yann, Lample Guillaume, and Polu Stanislas. Numinamath, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *NeurIPS*, 2022.

Gang Li, Yan Chen, Ming Lin, and Tianbao Yang. Drpo: Efficient reasoning via decoupled reward policy optimization. *arXiv preprint arXiv:2510.04474*, 2025a.

Gang Li, Ming Lin, Tomer Galanti, Zhengzhong Tu, and Tianbao Yang. Disco: Reinforcing large reasoning models with discriminative constrained optimization. In *NeurIPS*, 2026.

Jiaming Li, Longze Chen, Ze Gong, Yukun Chen, Lu Wang, Wanwei He, Run Luo, and Min Yang. Implicit actor critic coupling via a supervised learning framework for rlvr. *arXiv preprint arXiv:2509.02522*, 2025b.

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 *ICLR*, 2024.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL*, 2017.

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.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

Lei Pang, Jun Luo, and Ruinan Jin. Tic-grpo: Provable and efficient optimization for reinforcement learning from human feedback. *arXiv preprint arXiv:2508.02833*, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

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

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. In *arXiv preprint arXiv:2402.03300*, 2024.

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

Yining Wang, Jinman Zhao, Chuangxin Zhao, Shuhao Guan, Gerald Penn, and Shinan Liu. $lambda$-grpo: Unifying the grpo frameworks with learnable token preferences. *arXiv preprint arXiv:2510.06870*, 2025.

Junkang Wu, Kexin Huang, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Quantile advantage estimation for entropy-safe reasoning. *arXiv preprint arXiv:2509.22611*, 2025a.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, et al. It takes two: Your grpo is secretly dpo. *arXiv preprint arXiv:2510.00977*, 2025b.

Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*, 2025.

Fengkai Yang, Zherui Chen, Xiaohan Wang, Xiaodong Lu, Jiajun Chai, Guojun Yin, Wei Lin, Shuai Ma, Fuzhen Zhuang, Deqing Wang, et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *Advances in Neural Information Processing Systems*, 38:163245–163284, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. In *NeurIPS*, 2026.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Hong Dingqian, and Hui Xiong. Gvpo: Group variance policy optimization for large language model post-training. In *NeurIPS*, 2026.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

Haizhong Zheng, Yang Zhou, Brian Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *Advances in Neural Information Processing Systems*, 38:124321–124346, 2026.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *NAACL*, 2024.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. In *NeurIPS*, 2026.

A OPEN-ANSWER MATHEMATICAL REASONING: THE SMALL-SPURIOUS-ADVANTAGE CASE

We work out the case $p_{\mathrm{g}} \approx 0$ deferred from Section 3.2.

Setup. Consider a prompt q on which the policy attains per-sample success probability p_q, and recall the decomposition of Eq. (3),

$$p_q \;=\; p_{\mathrm{r}} \;+\; (1 - p_{\mathrm{r}}) \cdot p_{\mathrm{g}},$$

where p_{r} is the probability that the policy obtains the correct answer through reasoning, and p_{g} is the probability of obtaining the correct answer by any other route. For open-answer mathematical reasoning, the answer surface is large (numbers, fractions, symbolic expressions, tuples, intervals), and the prompt typically admits a single correct surface form; producing this form without reasoning is a $\ll 1$ event, so $p_{\mathrm{g}} \to 0$.

Effect on the within-group count. With $p_{\mathrm{g}} \to 0$, the per-prompt success probability reduces to $p_q \approx p_{\mathrm{r}}$, and the within-group count $n^+ \sim \mathrm{Binomial}(G, p_q)$ has $\mathbb{E}[n^+] = G p_q \approx G p_{\mathrm{r}}$ — the count is driven by the reasoning channel alone. Substituting into Eq. (2), the rare-correct magnitude

$$|\hat{A}^+| \;=\; \sqrt{n^-/n^+}$$

is large precisely when p_{r} is small, i.e., when the policy is reasoning through the prompt only rarely.

Connection to prior work. This is the case in which the closed-form weighting of Mroueh (2025),

$$\omega^+(p) = \sqrt{(1-p)/p}, \qquad \omega^-(p) = -\sqrt{p/(1-p)},$$

admits the adaptive contrastive interpretation: a rare-correct rollout receives a large weight because reasoning success on this prompt is genuinely rare. When $p_{\mathrm{g}} = 0$, the second term of Eq. (3) contributes nothing, and the magnitude assigned by Eq. (2) corresponds to a learning signal with no spurious advantage.

Bound on the residual spurious advantage. When p_{g} is small but not zero, the expected fraction of the magnitude attributable to reasoning is

$$\frac{p_{\mathrm{r}}}{p_q} \;=\; \frac{p_{\mathrm{r}}}{p_{\mathrm{r}} + (1 - p_{\mathrm{r}}) p_{\mathrm{g}}},$$

which is $\geq 1 - p_{\mathrm{g}}/p_q$. For open-answer mathematical reasoning the ratio is close to 1, and the spurious advantage is at the level of p_{g} at most. The MATH-7.5K analysis of Section 3.3 quantifies this residual: even on a nominally open-answer training set, p_{g} is not zero and the residual is measurable, though small relative to the bounded-answer cases of Section 3.3.

B $|\hat{A}_i|$ AS A PER-ROLLOUT LEARNING WEIGHT

Throughout the main paper we describe $|\hat{A}_i|$ as the per-rollout scalar weight on rollout i's gradient contribution to the policy update. Strictly, the GRPO policy gradient for rollout i (ignoring the KL term and in the unclipped regime where $w_{i,t} \approx 1$) factorizes as

$$\nabla_\theta \mathcal{L}_i \;=\; -\,\hat{A}_i \cdot \underbrace{\frac{1}{|\tau_i|} \sum_t \nabla_\theta \log \pi_\theta(\tau_{i,t})}_{\text{rollout-specific log-prob gradient}},$$

so the gradient norm contributed by rollout i factorizes as $|\hat{A}_i| \cdot \left\| \frac{1}{|\tau_i|} \sum_t \nabla_\theta \log \pi_\theta(\tau_{i,t}) \right\|$. The second factor depends on the rollout's tokens and the current policy, but is independent of the choice of advantage estimator. For a fixed rollout, $|\hat{A}_i|$ therefore acts as a multiplicative scalar that determines how strongly that rollout's log-probability is pushed.

This usage matches the standard convention in the RL-for-LLM literature, where $|\hat{A}_i|$ is treated as the per-rollout learning weight when comparing advantage estimators (Schulman et al., 2017;

Ahmadian et al., 2024; Hu, 2026; Shao et al., 2024; Mroueh, 2025). In particular, Mroueh (2025) analyzes GRPO at this per-rollout weight level under binary rewards and derives the same $\sqrt{n^-/n^+}$ form we use in Eq. (2). Our analysis of the spurious advantage component is therefore a statement about how this per-rollout weight is composed across reasoning-driven and non-reasoning success channels, not a claim about gradient norms in absolute units or about the precise effect under specific PPO clip realizations.

C Analysis of the harness

This appendix proves the statements of Section 3.5 and records the decomposition check used in Section 4.3. Throughout, $G = 16$, $b = 5$, rewards are binary, and a group with $n^+ \in \{0, G\}$ is dropped before the loss (Appendix E).

Lemma 1 (Advantage identities)**.** *For the estimator of Eq. (1) with $\varepsilon = 0$ and $0 < n^+ < G$, $\sum_i \hat{A}_i = 0$, $\sum_{i:\, r_i=1} \hat{A}_i = \sqrt{n^+ n^-}$ and $\sum_i |\hat{A}_i| = 2\sqrt{n^+ n^-}$. All three vanish when $n^+ \in \{0, G\}$.*

Proof. By Eq. (2), each of the n^+ correct rollouts carries $\sqrt{n^-/n^+}$ and each of the n^- wrong rollouts carries $-\sqrt{n^+/n^-}$. Summing gives $n^+\sqrt{n^-/n^+} = \sqrt{n^+ n^-}$ on the positive side and $-\sqrt{n^+ n^-}$ on the negative side. With $\varepsilon > 0$ the identities hold up to the factor $\sigma/(\sigma + \varepsilon)$. The implementation uses the unbiased standard deviation, which multiplies every advantage by $\sqrt{(G-1)/G} = 0.968$ for $G = 16$ and leaves every ratio in this paper unchanged. □

Lemma 2 (Binomial sandwich)**.** *Let $n^+ \sim \mathrm{Binomial}(G, p)$ and $z(p) = p^G + (1-p)^G$. Then $m(p) = \mathbb{E}\sqrt{n^+ n^-} = (1 - z(p))\, c(p)$ with $c(p) = \mathbb{E}[\sqrt{n^+ n^-} \mid 0 < n^+ < G] \in [\sqrt{G-1},\, G/2]$, and $m(p) \le \sqrt{G(G-1)p(1-p)}$.*

Proof. The factorization is the law of total expectation, since $\sqrt{n^+ n^-} = 0$ on the event $n^+ \in \{0, G\}$. For $1 \le k \le G-1$, $k(G-k)$ is smallest at $k \in \{1, G-1\}$ with value $G-1$ and largest at $k = G/2$ with value $G^2/4$, which gives the range of $c(p)$. For the upper bound, $\mathbb{E}[n^+ n^-] = G(G-1)p(1-p)$, and Jensen's inequality for the concave square root gives $m(p) \le \sqrt{G(G-1)p(1-p)}$. For $G = 16$, $m(0.05) = 2.48$ with $1 - z(0.05) = 0.56$, and $m(0.5) = 7.74$ against the bound 7.75. □

Proof of Proposition 1. Selection reads q only, so given a kept prompt the G rollouts are drawn from $\pi_{\theta_{\text{old}}}(\cdot \mid q)$ exactly as under GRPO, and every function of the group, including $z(q)$, $\bar{g}(q)$ and the identities of Lemma 1, is unchanged. The prompt marginal becomes $\mathcal{D}_H(q) = w(q)\,\mathcal{D}(q)$ with $w(q) = M\,P(q \text{ kept} \mid q \in \mathcal{C}_t) \in [0, M]$, which depends on the prior and on M but not on θ. With the binary offline prior on the 0.5B model, 63.5% of the prompts are unflagged; an unflagged candidate is kept with probability close to $b/(0.635\,Mb)$, so $w \approx 1.6$, and a flagged candidate is kept only when fewer than b of the Mb candidates are unflagged, so $w \approx 0.01$. For the update, Eq. (7) averages the per-rollout terms of the N_t^{grp} kept groups, so $\hat{g}_t$ is the mean of N_t^{grp} group terms whose conditional mean does not depend on N_t^{grp} and whose between-group variance we write as σ_g^2; hence $\mathrm{Var}(\hat{g}_t \mid N_t^{\text{grp}}) = \sigma_g^2 / N_t^{\text{grp}}$. By linearity, $\mathbb{E}[N_t^{\text{grp}}] = b(1-\bar{z})$ with $\bar{z}$ the zero-gradient rate of the drawn groups. Under GRPO, $N_t^{\text{grp}} \sim \mathrm{Binomial}(b, 1-\bar{z})$; under the harness the kept prompts are exchangeable but dependent, so the binomial law is an approximation while the mean is exact. The ratio of expected trainable groups at equal rollouts is therefore $(1-\bar{z}_H)/(1-\bar{z}_{\text{GRPO}})$, and to first order, $\mathbb{E}[1/N_t^{\text{grp}}] \approx 1/\mathbb{E}[N_t^{\text{grp}}]$, the same factor applies to the noise term. An update without a trainable group changes no parameter, which happens with probability $\bar{z}^b$ (0.6% under GRPO and 0.02% under the harness on the 0.5B model). Conditioning on $N_t^{\text{grp}} \ge 1$ and using the binomial law, $\mathbb{E}[1/N_t^{\text{grp}}]$ is 0.363 under GRPO and 0.259 under the harness, a ratio of 1.40 against 1.29 to first order. □

Remark 1 (Relation to the non-convex SGD bound). For an L-smooth objective, a step size $\eta \le 1/L$ and stochastic gradients with variance at most σ^2, the standard bound (Ghadimi and Lan, 2013) reads $\min_{t \le T} \mathbb{E}\|\nabla J(\theta_t)\|^2 \le 2\,(J(\theta_0) - J^\star)/(\eta T) + \eta L \sigma^2$. Applied to each method with its own prompt marginal and its own trainable-conditional gradient, the variance term at equal η and equal rollouts scales with $\mathbb{E}[1/N_t^{\text{grp}}]$, so it is smaller under the harness by the factor of Proposition 1. Two

objectives are involved, the expectations of the same per-prompt objective under $\mathcal{D}$ and under $\mathcal{D}_H$, so this is a statement about the per-update estimator and not a comparison of convergence rates or of maximizers. The KL term and the length normalization of the sequence mean are further sources of bias that the bound ignores. Clipping is inactive in our runs, since one optimizer step follows each generated batch and the ratio equals one up to numerical mismatch (Appendix E).

Corollary 1 (The gain is set by the predictive value of the score)**.** *Write* $1-\bar{z}_H = \mathbb{E}_{q\sim\mathcal{D}_H}[1-z(q)]$. *If the score is independent of* $z(q)$*, then* z *has the same distribution under* $\mathcal{D}_H$ *as under* $\mathcal{D}$ *and the factor of Proposition 1 equals one; this is the shuffled control (*0.99 *on the 0.5B model, Table 9). If the score was predictive of* $z(q)$ *under the policy that produced the pass but is not under the current policy, the factor again tends to one; this is the stale prior on the 3B model, where the pass flags* 6.3% *of the prompts and covers* 9.4% *of the zero-gradient groups (Table 16) and the factor is* 1.01*. Re-estimating the score under a later policy restores it, to* 1.33 *in the trainable share and* 1.37 *in mass. This is the reason for the refresh rule of Section 3.5.*

Remark 2 (The spurious share is invariant to selection). The share s of Eq. (8) depends on the distribution $H(k)$ of n^+ over trainable groups and on the flawed rates $f(k)$. Selection changes $H(k)$ but not $f(k)$, and $f(k)$ is nearly flat for $k \geq 2$ (Table 17), so s moves little (42.3% under GRPO and 41.8% under the harness, Table 6). The harness therefore changes how many groups carry an advantage, not the fraction of that advantage that is spurious, which is what Section 4.3 reports.

Decomposition check. Table 9 splits the positive advantage mass per drawn group, $\mathbb{E}[\sqrt{n^+n^-}]$ over all drawn groups, into the trainable share $1-\bar{z}$ and the mean magnitude c of a trainable group, with $H(k)$ and $\bar{z}$ read from the training logs of every run. The mass ratio to GRPO is the product of the two ratios. On the 0.5B model the harness raises the trainable share by 1.29 and the mean magnitude by 1.03, so the 1.33 gain in mass is a count effect, as Proposition 1 states, and the shuffled control changes neither factor. The $f(k)$-weighted spurious mass of Table 6 (1.31) differs from the mass ratio by the ratio of the spurious shares. The last column gives $\mathbb{E}[1/N_t^{\mathrm{grp}}]$ under the binomial law conditioned on at least one trainable group.

Model	Setting	$1-\bar{z}$	c	Mass	$1-\bar{z}$ rel.	c rel.	Mass rel.	$\mathbb{E}[1/N_t^{\mathrm{grp}}]$
0.5B	GRPO	0.639	6.16	3.93	1.00	1.00	1.00	0.363
	Shuffled control	0.634	6.09	3.86	0.99	0.99	0.98	0.366
	Level-label prior	0.767	6.25	4.79	1.20	1.01	1.22	0.284
	Online blend	0.850	6.38	5.42	1.33	1.04	1.38	0.247
	Ours, stale prior	0.822	6.37	5.23	1.29	1.03	1.33	0.259
3B	GRPO	0.520	6.01	3.12	1.00	1.00	1.00	0.458
	Shuffled control, stale prior	0.516	6.04	3.12	0.99	1.01	1.00	0.463
	Shuffled control, refreshed prior	0.540	6.04	3.27	1.04	1.01	1.04	0.441
	Online blend	0.614	6.11	3.75	1.18	1.02	1.20	0.381
	Ours, stale prior	0.528	6.02	3.18	1.01	1.00	1.02	0.452
	Ours, refreshed prior	0.691	6.19	4.27	1.33	1.03	1.37	0.327

Table 9: Decomposition of the positive advantage mass per drawn group into the trainable share $1-\bar{z}$ and the mean magnitude c of a trainable group, from the training logs (seeds as in Table 15). Rel. columns are ratios to GRPO of the same size. We shade the proposed method.

D JUSTIFICATION OF THE MATH-7.5K ANSWER-SHAPE CATEGORIZATION

A nominally open-answer training set can also exhibit the bounded case on a sizeable subset of its problems. We analyze MATH-7.5K by the syntactic shape of the gold answer, and group them into 11 shape categories (Table 10; the per-quantity derivations follow). Five of the 11 categories are flagged as *bounded*, together covering 55.95% of MATH-7.5K, indicating that even an open-answer benchmark can contain a sizeable subset with closed answer form. Figure 5 gives the per-string guess baseline of this training set and Figure 6 the per-level breakdown, where the bounded share is highest on the easiest problems.

This appendix walks through every quantity in Table 10 column by column, so each cell can be re-derived from the underlying definitions.

Category	# / Share (%)	Guess hit rate (p_g)	Bounded
`int_small` [-10,10]	1806 / 24.41	4.76%	✓
`int_medium` [-100,100]	1701 / 22.99	0.50%	✓
`expression`	1544 / 20.87	0.00%	–
`int_large`	1206 / 16.30	0.00%	–
`simple_fraction`	556 / 7.51	1.00%	✓
`tuple_or_list`	270 / 3.65	0.00%	–
`other`	121 / 1.64	0.00%	–
`decimal_short`	116 / 1.57	0.00%	–
`finite_set_listed`	46 / 0.62	20.00%	✓
`percent`	31 / 0.42	0.99%	✓
`empty`	2 / 0.03	0.00%	–

Table 10: Answer-shape distribution of MATH-7.5K over the 7,399 parsable problems. Guess hit rate: probability that a random answer of the shape matches the gold answer ($1/S$ for a finite surface of size S, 0 otherwise). The five bounded categories cover 55.95% of MATH-7.5K.

Source set (the denominator of every column). We work with the standard MATH training split of 7,500 problems. We extract the gold answer of each problem by parsing the `\boxed{...}` token in the official solution. 7,399 problems contain a parsable `\boxed{}`; the remaining 101 are dropped where parsing fails. Every column of Table 10 is computed over this set of 7,399 problems.

Category (the row label). Each parsable gold answer is assigned to one of 11 surface-syntactic categories, defined purely by the shape of the answer string rather than the underlying problem domain. For example, an answer of 5 goes to `int_small` regardless of whether the problem is about geometry or arithmetic; an answer of p/q with $|p|, |q| \leq 20$ goes to `simple_fraction`; a non-numeric algebraic answer such as x^2+1 goes to `expression`. The 11 categories are mutually exclusive and exhaustive, with `other` as a catch-all.

#Problems (column 2). The count of problems in the source set whose gold answer falls in this category. The 11 counts sum to 7,399.

Share (column 3). #Problems divided by 7,399, expressed as a percent. The 11 shares sum to 100%. This column tells the reader how large each shape category is relative to the whole training set.

Guess hit rate (column 4). The probability that a policy producing a uniformly random answer of the given syntactic shape lands on the gold answer. Two cases:

- *Finite surface.* If the category has a finite answer surface of size S, the hit rate is $1/S$. For `int_small` (S=21 integers in $[-10, 10]$), this gives $1/21 \approx 4.76\%$; for `int_medium` (S=201), $1/201 \approx 0.50\%$; for `simple_fraction` ($S\approx$100 irreducible fractions with $|p|, |q| \leq 20$), $\approx 1.00\%$; for `finite_set_listed` (typical S=5 enumerated options), $\approx 20.00\%$; for `percent` (S=101 percent values), $\approx 0.99\%$.
- *Open-ended surface.* If the category has an infinite or open-ended answer surface (e.g., `expression`, `int_large`, `tuple_or_list`, `decimal_short`, `other`, `empty`), the hit rate is treated as 0 — emitting a random expression or large integer that exactly matches the gold answer has negligible probability.

This column is the per-category estimate of p_g in Eq. (3): the probability of obtaining the correct answer without reasoning.

Bounded (column 5). A binary flag set to ✓ when the guess hit rate is $\geq 0.5\%$, and − otherwise. The threshold is conservative: we only flag categories where a no-reasoning policy already has a measurable hit rate, not categories with a marginal rate that would be hard to distinguish from sampling noise. Under this rule, 5 of the 11 categories are flagged (`int_small`, `int_medium`, `simple_fraction`, `finite_set_listed`, `percent`), together covering 4,140 problems (55.95% of MATH-7.5K).

Why this analysis is defensible. Three points clarify the reading of Table 10: (i) the categorization is syntactic, not semantic, since we group answers by surface shape, not by problem difficulty, so the table reports a fact about the answer surface of MATH-7.5K rather than a re-interpretation of the benchmark; (ii) the 0.5% threshold flags only categories with a measurable hit rate, and the largest bounded category, `int_small` (24.41% of the data), has a guess rate of 4.76%, well above the threshold; (iii) we do not claim that bounded problems in MATH-7.5K are easy or unimportant — we claim only that the bounded case of Section 3.3 arises on a sizeable subset of MATH-7.5K, so the spurious advantage component is non-zero even on a benchmark conventionally treated as open-answer. Section 3.3 provides further empirical evidence.

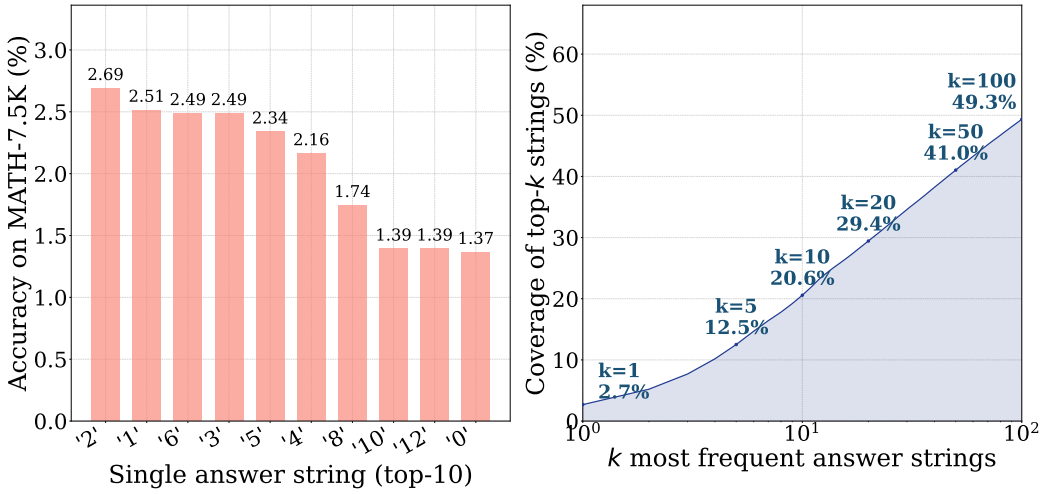

Figure 5: A policy that does no reasoning at all can still score on MATH-7.5K just by guessing popular answer strings. **Left**: per-string accuracy of a constant policy that always emits one fixed string from the top-10 list (emitting "2" alone scores 2.69%). **Right**: coverage of the gold answers by the k most frequent answer strings (20.6% at k=10, 49.3% at k=100). Even on a free-form math benchmark, there is a measurable floor from guessing.

Figure 6 breaks down the bounded share of Table 10 by subject and by difficulty level.

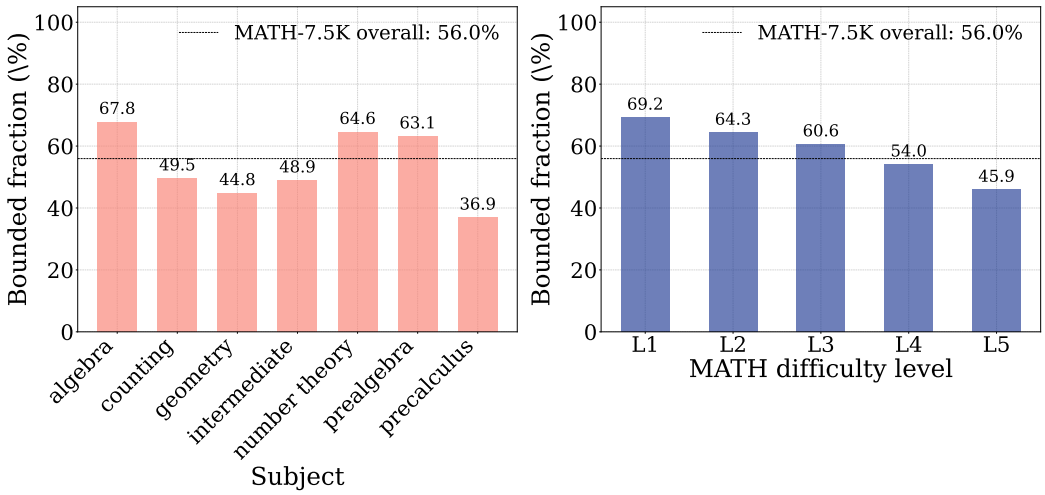

Figure 6: Bounded-answer share of MATH-7.5K by subject (left) and by MATH difficulty level (right), over the 7,399 parsable problems. The share decreases from 69.2% at Level 1 to 45.9% at Level 5.

E Implementation Details

Trainer. All settings share one trainer, which differs across settings only in the prompt-selection layer and, for the ablations, in the advantage estimator. Specifically, the policy is Qwen2.5-0.5B-Instruct or Qwen2.5-3B-Instruct, held in fp32 for optimization and exported in bf16 to the sampler after every update. Each update uses $b = 5$ prompts with $G = 16$ rollouts each, sampled by vLLM at temperature 1.0 (top-p 1.0, no repetition penalty) with at most $8{,}192$ new tokens per rollout. Rollouts cut at the token limit remain in the batch. The loss is the clipped PPO-style surrogate of GRPO with the advantage of Eq. (1), with $\epsilon = 0.2$ on both sides and the sampling-time log-probabilities as $\pi_{\theta_{\text{old}}}$. The aggregation is a sequence mean (the token mean of each rollout, then the mean over the rollouts of the update). The standard deviation of Eq. (1) is the unbiased estimate in the implementation, which multiplies every advantage by $\sqrt{(G-1)/G} = 0.968$ for $G = 16$ and leaves every ratio in the paper unchanged. The optimizer is AdamW with learning rate 10^{-6}, no weight decay, $(\beta_1, \beta_2) = (0.9, 0.999)$ and gradient-norm clipping at 1.0. One optimizer update follows each generation batch, so training is on-policy. A run has 200 updates with a checkpoint every 20, and gradient checkpointing is enabled at 3B. Note that a group whose rewards are all equal (all-wrong or all-correct) is dropped before the advantage is computed. This rule is applied for every estimator, including the reweighting estimator, so the all-correct branch of Algorithm 3 never reaches the optimizer. An update whose five groups are all dropped changes no parameter.

Training data and epoch stream. The training set is the MATH training split (Hendrycks et al., 2021), restricted to the $7{,}496$ problems whose gold answer is enclosed in a `boxed` command. Table 10 uses a separate answer parser, which accepts $7{,}399$ problems of the same split. The seed fixes a random order of the training prompts (the epoch stream), which every setting reads from the same position. GRPO takes $b = 5$ prompts per update, $1{,}000$ per run. The harness and its shuffled control take $Mb = 15$ candidates per update, $3{,}000$ per run. Thus, no setting reaches the end of the stream. The reward is binary, 1 when the boxed answer of the rollout matches the gold answer after normalization and 0 otherwise. The same system prompt, temperature and reward are used in training, in the offline pass and in the derivation-check rollouts.

Prior construction. The lookup prior $\hat{P}(q)$ is built from one offline pass of the initial policy over the training prompts. Specifically, the pass samples $G = 16$ rollouts per prompt with the training system prompt, temperature 1.0 and at most $4{,}096$ new tokens, and scores them with the binary reward. It sets $\hat{P}(q) = 1$ when the group is all-wrong or all-correct and $\hat{P}(q) = 0$ otherwise. At 0.5B the pass covers all 7,496 prompts. At 3B it covers a random subset, 3,878 prompts for the stale prior and 2,500 for the refreshed prior. A prompt outside the subset receives the mean $\hat{P}$ of its MATH difficulty level over the covered prompts. A level with fewer than 20 covered prompts uses the overall mean instead. A third table, `prior_level.json`, assigns the score $\ell/5$ from the MATH difficulty label $\ell \in \{1, \ldots, 5\}$ of each prompt and needs no offline pass. It is used only for the level-label control of Appendix F, together with a permuted copy for its shuffled counterpart. The prior is an input of a run, not a quantity the run updates. It is built offline either from the initial policy or from the policy after 40 GRPO updates. The former is the stale prior. The latter, i.e., the seed-1 GRPO checkpoint at update 40 of the same model size, is the refreshed prior. Both use the same pass and the same rule, and the chosen table is used throughout the run. The rule for choosing between the two is stated in Section 3.5. A prompt counts as flagged when its prior score exceeds 0.5. For the stale prior these are the prompts with $\hat{P}(q) = 1$, and for the refreshed prior they include the prompts whose level mean exceeds 0.5.

Selection rule. With $M = 3$, each update takes $Mb = 15$ candidates $\mathcal{C}_t$ from the stream, scores each candidate by $s(q) = \hat{P}(q)$, and keeps the $b = 5$ candidates with the smallest score as $\mathcal{S}_t$. Note that ties are frequent because the stale prior is binary, and they are broken by a seeded uniform draw per candidate. The ten candidates that are not selected are removed from the stream and not returned. Two options are available for ablations and are off in the main settings. The first, additive noise, adds $u \sim \mathcal{U}(0, \varepsilon_{\text{s}})$ to $s(q)$ before ranking. It lets a flagged candidate be selected when its noisy score is below that of an unflagged candidate. The second, the online curriculum blend, replaces $s(q)$ by $\frac{1}{2}s(q) + \frac{1}{2}d_{\ell(q)}$. The term d_ℓ is an exponential moving average (rate 0.1, initial value 0.5) of the indicator that groups of difficulty level ℓ were zero-gradient under the current policy.

Shuffled control. The shuffled control reads a second table with the same format, in which the multiset of prior values is permuted at random across prompts under a fixed seed. All other details, including M, the tie-breaking draws and the removal of unselected candidates, remain the same as in the harness setting.

DAPO setting. This setting adds the dynamic sampling of DAPO (Yu et al., 2026) to the GRPO trainer, and the rest of the trainer remains the same. Specifically, after the b groups of an update are generated, the trainer counts the groups that are not zero-gradient. When fewer than q remain, it draws further prompts from the stream, generates their groups and repeats, up to three generation rounds per update. Thus, each update samples more than bG rollouts, and a run with 200 updates would exceed the 16,000 rollouts of the other settings. DAPO therefore stops at 136 updates, which samples 16,144, 15,952 and 16,560 rollouts at 0.5B for seeds 1–3, and 18,896 at 3B (seed 1). Its reported checkpoint is the one at update 136.

Reweighting setting. The ablation replaces Eq. (1) by Algorithm 3 with $c = 1$, so a correct rollout receives $+1$ and a wrong rollout receives $-n^+/n^-$. The trainer, the clip, the KL term and the loss aggregation are those of the GRPO setting. Zero-gradient groups are dropped before the estimator is applied, as in every setting.

Evaluation. All numbers in the paper come from the final checkpoint of a run (update 200, or 136 for DAPO). The checkpoint is never chosen on a test set. The evaluation covers 20 benchmarks with one engine configuration (bf16, engine seed 1,234, context 8,192, at most 2,048 new tokens). Greedy decoding is used on GSM8K (1,319 problems), MATH-500 (500), Minerva-Math (272), OlympiadBench (566, text-only numerical subset), MMLU-Math (748, 5-shot) and MAWPS (520). It is also used on ASDiv (1,218), SVAMP (300), College-Math (2,818), TheoremQA (747), MMLU-Pro (1,000-problem subset, 10-option, 5-shot), ARC-Challenge (1,172, 5-shot) and Omni-MATH (1,000-problem subset). The small competition and multiple-choice sets are sampled 16 times at temperature 0.6 and top-p 0.95, and their score is the mean accuracy over the 16 samples (avg@16). These sets are SAT-Math (32, 5-shot), AQuA (254, 4-shot), the AQuA validation split (254), AMC 2022–2023 (83), AIME 2022–2024 (90), AIME 2024 (30) and AIME 2025-I (15). Avg8 is the mean over the eight core benchmarks GSM8K, MATH-500, Minerva-Math, OlympiadBench, MMLU-Math, SAT-Math, AQuA and AMC. The ten remaining benchmarks are reported in Table 12 (Appendix F); the AQuA validation split and the AIME 2022–2024 set duplicate the AQuA and AIME families at other splits and are not tabulated.

Seeds. A seed fixes the epoch stream, the sampler stream and the tie-breaking draws of the selection rule. At 0.5B, GRPO, the harness, the shuffled control, the online blend and DAPO are evaluated on seeds 1–3 (Table 15). At 3B, GRPO and the harness with the stale prior are evaluated on seeds 1–3, the shuffled control with the stale prior on seeds 1–2, the online blend on seeds 1–2 and DAPO on seed 1. The harness and the shuffled control with the refreshed prior are evaluated on seeds 1–6 (Table 13).

Hardware. One training run occupies two NVIDIA A100 80GB GPUs, one for the optimizer and one shared by the vLLM sampler and the reference-log-probability server. The offline pass and the evaluation of a checkpoint each use one GPU.

F Additional Diagnostics

This appendix collects the diagnostics that support Section 4: the random-guess floor of the bounded-answer benchmarks; the per-seed results at 0.5B (Table 11), the n^+ distribution (Figure 7) and the ten benchmarks outside the core eight (Table 12); the per-size mechanism table (Table 14); the full rollout budget (Table 15); the coverage table (Table 16); the parsing rules of the derivation check and its bucketed version (Table 17); the level-label filter control; and the prior-based calibration arm (Table 20).

Random-guess share of correct rollouts. On each benchmark, we quantify how much of a correct rollout could already be produced by a random-guess policy. For a k-option multi-choice bench-

mark, a uniform random-guess policy attains accuracy $\geq 1/k$. Under a first-order uniform-guess assumption, the random-guess share of correct rollouts is $\frac{(1/k)(1-s)}{\text{acc}}$, where $s = (\text{acc}-1/k)/(1-1/k)$. On the untrained Qwen2.5-0.5B-Instruct, this share is 53.4% on AQuA (k=5), 30.1% on MMLU-math (k=4), and 22.8% on SAT-Math (k=4); on the open-answer math benchmarks (MATH-500, GSM8K, Minerva-Math, Olympiad), the random-guess floor is 0 and the share is 0. The bounded-answer benchmarks therefore feed the estimator a within-group composition that is partially random-guess even at the start of training.

Seed noise. Table 2 reports, for the three GRPO seeds at 0.5B, the mean, the standard deviation and 2×SEM of every core benchmark at the final checkpoint. The Avg8 standard deviation over seeds is 1.14. The per-benchmark 2×SEM ranges from 0.53 on GSM8K and 1.40 on MATH-500 to 4.45 on MMLU-Math. Thus, a single-benchmark difference of a few points between two settings lies within the seed noise on the multiple-choice sets. The paired differences of Section 4 are compared with these values (Table 2 in Section 4.1).

Table 11 lists the per-seed Avg8 of the 0.5B settings, Table 13 those of the 3B pairs, and Figure 7 the distribution of n^+ over the trainable groups of five settings.

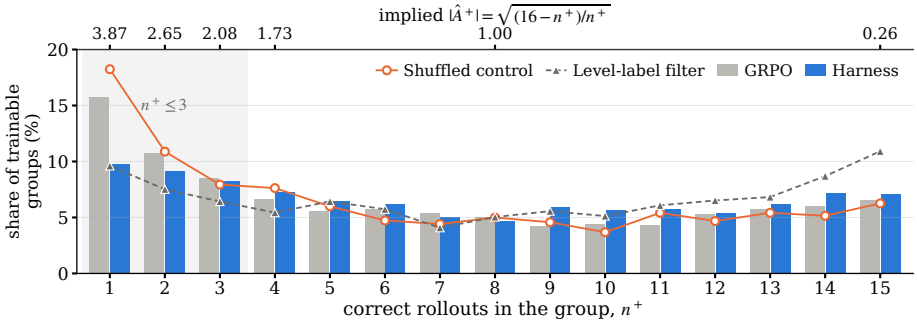

Figure 7: Distribution of n^+ over the trainable groups ($0 < n^+ < 16$), pooled over three seeds and 200 updates on the 0.5B model, for GRPO, the harness, the shuffled control and the level-label filter; the shaded region marks $n^+ \leq 3$ and the top axis gives the implied $|\hat{A}^+|$.

Seed	GRPO	Ours	Shuffled	Level	Reweight.	Blend
1	34.17	35.76	34.49	34.74	33.37	35.60
2	35.22	35.54	33.06	34.76	34.06	34.64
3	32.93	35.07	34.18	34.47	34.04	34.18
Mean±SD	34.11±1.14	35.46±0.35	33.91±0.75	34.66±0.17	33.83±0.39	34.80±0.73
Ours−col (paired)	–	–	+1.54±0.83	+0.80±0.21	+1.63±0.69	+0.65±0.43
p (two-sided)	–	–	0.084	0.022	0.055	0.118

Table 11: Per-seed Avg8 of the settings on the 0.5B model, with the paired difference harness minus column on the same seed and its two-sided paired t-test p.

The ten benchmarks outside the core eight. Table 12 reports the same runs on the ten remaining benchmarks. On these sets the three settings stay within seed noise of one another at both sizes, so the paper draws no conclusion from them.

Rollout budgets. Table 15 lists the number of rollouts sampled by every run, summed over its updates from the training log. Note that GRPO, the harness, the shuffled control and the online blend sample 16,000 rollouts for every seed and every size, since selection reads the lookup prior without additional rollouts. The runs with dynamic sampling are the exceptions listed in Appendix E. Specifically, they sample 23,568 rollouts for 200 updates at 0.5B, and 16,144, 15,952, 16,560 (0.5B, seeds 1–3) and 18,896 (3B, seed 1) for the equal-compute stop at 136 updates. The table also lists the offline passes that build the priors, 119,936 rollouts at 0.5B and 62,048 and 40,000 at 3B for the stale and the refreshed prior. The column "incl. prior" amortizes a pass over the evaluated runs that

Model	Setting	MAWPS↑	ASDiv↑	SVAMP↑	College-Math↑	TheoremQA↑	MMLU-Pro↑	ARC-C↑	AIME24↑	AIME25↑	Omni-MATH↑	Avg10↑
0.5B	GRPO	67.31	88.10	61.11	34.85	16.24	13.63	47.84	0.35	0.14	4.63	33.42±0.45
	Shuffled control, stale prior	66.28	87.99	61.00	34.97	17.09	13.00	49.37	0.28	0.14	4.87	33.50±0.25
	Ours, stale prior	66.86	88.29	62.67	34.50	19.46	14.00	47.64	0.35	0.14	5.50	33.94±0.20
3B	GRPO	68.27	96.36	90.78	52.76	35.34	37.30	75.43	5.00	2.92	13.27	47.74±0.57
	Shuffled control, stale prior	68.65	97.21	90.17	52.50	35.54	35.75	79.31	5.42	3.33	13.25	48.11±0.53
	Shuffled control, refreshed prior	67.37	96.24	90.83	52.60	33.40	35.98	81.77	6.11	3.68	13.50	48.15±1.00
	Ours, stale prior	68.46	96.72	90.00	52.80	34.61	38.80	83.32	6.04	3.96	12.65	48.74±1.06
	Ours, refreshed prior	68.65	95.99	89.67	52.50	34.67	34.58	80.55	6.60	3.33	13.48	48.00±1.15

Table 12: Accuracy (%, ↑) on the ten benchmarks outside the core eight and their mean (Avg10), for the settings with breadth evaluations. On these sets GRPO, the shuffled control and Ours stay within seed noise of one another at both sizes; the core-eight results are in Tables 3 and 8.

Seed	Ours (refr.)	Shuffled (refr.)	Δ	Ours (stale)	Shuffled (stale)	Δ
1	54.06	52.25	+1.82	55.19	48.62	+6.57
2	56.94	59.45	-2.51	58.32	51.39	+6.92
3	50.25	48.50	+1.76	55.74	–	–
4	50.92	55.55	-4.63	–	–	–
5	50.28	51.46	-1.19	–	–	–
6	56.11	54.00	+2.11	–	–	–
Mean±SD	53.09±3.02	53.54±3.76	-0.44±2.79	56.41±1.67	50.01±1.96	+6.75±0.25

Table 13: Avg8 per seed on the 3B model for the refreshed pair (six seeds) and the stale pair (three and two seeds), with the paired difference Δ.

use the prior built from it. Specifically, the rollouts of the pass are divided equally among these runs and added to the budget of each run. This gives 29,326 rollouts per 0.5B run with the stale prior (nine evaluated runs share the pass) and 19,333 per 3B run with the refreshed prior (twelve runs).

Derivation check. The derivation check of Section 3.3 is a deterministic rule applied to correct rollouts on multiple-choice prompts. It is applied only to prompts whose options all carry distinct numeric values. Specifically, for a rollout, the check locates the final answer statement of the chosen letter, e.g., a boxed letter, "the answer is" or a parenthesized letter. It then collects every number written before that position. When no number is found, the rollout is labeled "no derivation". When the last number equals the value of the chosen option (relative tolerance 10^{-6}), the rollout is labeled "supported". In the remaining case the rollout is labeled "derivation mismatch", whether the last number equals another option or none. A prompt whose chosen option has no numeric value is labeled "uncertain" and left out. Table 17 reports the share of derivation mismatch and no derivation among the correct rollouts of each n^+ bucket. The rollouts come from $8{,}000$ AQuA-RAT training prompts with 16 rollouts each, before training and after 200 GRPO updates on the same prompts.

Bucketed derivation check and spurious share. Table 17 is the bucketed version of the derivation check of Section 3.3; the buckets are not prompt-matched, so the prompt-matched Table 1 is the one used in the main text. Table 18 reports the spurious share of the positive weight of Eq. (8) for every setting at both sizes.

Derivation check on trained policies. Table 5 (Section 4.3) applies the same check to trained policies. The rollouts are drawn on one subset of 1,500 AQuA-RAT training prompts with 16 rollouts per prompt. The policies are the untrained model and the final checkpoints of GRPO, the harness and the shuffled control at 0.5B (three seeds each). The table reports the flawed share, i.e., derivation mismatch plus no derivation, among the correct rollouts of each policy, over all correct rollouts and for the buckets $k = 1$ and $k \geq 8$. Its last column weights each correct rollout by the advantage magnitude of its group. Specifically, it reports $\sum_i |\hat{A}^+(k_i)|\, f_i / \sum_i |\hat{A}^+(k_i)|$ over the correct rollouts of groups with $0 < k < 16$. The term k_i is the number of correct rollouts in the group of rollout i, and $f_i \in \{0, 1\}$ marks a flawed rollout. The weighted flawed share is 57.9% for the untrained model, 54.9% for GRPO, 51.1% for the harness and 54.1% for the shuffled control (three-seed means).

Validation of the derivation check. The derivation check is a fixed rule, so we validated its labels against an LLM judge that reads the full rollout under a fixed rubric. Specifically, 100 correct rollouts were sampled with a fixed seed from the untrained, GRPO, harness and shuffled-control dumps of seed 1. The sample is stratified by checker label (34 supported, 34 derivation mismatch and all 32 no derivation), and the judge did not see the checker label. Note that the reference labels

Model	Setting	Zero-gradient rate↓	Mean $\lvert A^+\rvert \downarrow$	$n^+ \leq 3 \downarrow$	n
0.5B	GRPO	36.1%±1.2%	1.655±0.006	35.1%±0.7%	3
	Level-label filter	23.3%±2.1%	1.326±0.055	23.6%±1.9%	3
	Shuffled, stale prior	36.6%±1.9%	1.735±0.036	37.0%±1.6%	3
	Ours (stale prior)	17.8%±1.7%	1.438±0.051	27.1%±1.0%	3
3B	GRPO	48.0%±0.8%	1.100±0.044	16.9%±1.2%	3
	Shuffled, stale prior	48.4%±1.3%	1.154±0.038	19.9%±1.2%	2
	Shuffled, refreshed prior	46.0%±2.4%	1.115±0.035	17.7%±0.9%	6
	Ours (stale prior)	47.0%±0.6%	0.918±0.014	11.2%±2.1%	2
	Ours (refreshed prior)	30.9%±1.0%	1.094±0.033	16.6%±1.7%	6

Table 14: Zero-gradient rate over training, mean $|\hat{A}^+|$ over trainable groups ($0 < n^+ < G$), and share of trainable groups with $n^+ \leq 3$, for each setting on the 0.5B and 3B models. Cells report mean ± standard deviation over n seeds. Arrows give the direction the harness aims at (↓ lower). We shade the proposed method, on the 3B model with the prior of the initial policy and with the refreshed prior.

Model	Setting	Updates	Rollouts sampled per seed↓	Mean↓	Incl. prior (per run)↓	n
0.5B	GRPO	200	16,000/16,000/16,000	16,000	16,000	3
	DAPO	136	16,144/15,952/16,560	16,219	16,219	3
	Reweighting	200	16,000/16,000/16,000	16,000	16,000	3
	Level-label filter	200	16,000/16,000/16,000	16,000	16,000	3
	Online curriculum, stale prior	200	16,000/16,000/16,000	16,000	29,326	3
	Shuffled control, stale prior	200	16,000/16,000/16,000	16,000	29,326	3
	Ours, stale prior	200	16,000/16,000/16,000	16,000	29,326	3
	Offline pass for the stale prior (7,496 prompts × 16)	0	–	119,936	–	–
3B	GRPO	200	16,000/16,000/16,000	16,000	16,000	3
	DAPO	136	18,896	18,896	18,896	1
	Online curriculum, stale prior	200	16,000/16,000	16,000	26,341	2
	Shuffled control, stale prior	200	16,000/16,000	16,000	26,341	2
	Shuffled control, refreshed prior	200	16,000/16,000/16,000/16,000/16,000/16,000	16,000	19,333	6
	Ours, stale prior	200	16,000/16,000	16,000	26,341	2
	Ours, refreshed prior	200	16,000/16,000/16,000/16,000/16,000/16,000	16,000	19,333	6
	Offline pass for the stale prior (3,878 prompts × 16)	0	–	62,048	–	–
	Offline pass for the refreshed prior (2,500 prompts × 16)	0	–	40,000	–	–

Table 15: Rollouts sampled per run (sum over updates), by setting, size and seed.

come from this LLM judge and not from human annotators, and that the sample is label-stratified, not population-representative. Table 19 reports the agreement per label and the precision and recall of the checker's "flawed" labels (derivation mismatch or no derivation) with the judge as reference. The agreement is 81% overall (76.5% for supported, 88.2% for derivation mismatch and 78.1% for no derivation), the precision of "flawed" is 0.833 and its recall 0.873, and Cohen's κ is 0.586. The disagreements show two error modes. First, the checker marks a rollout as supported when the value of the chosen option is restated after a chain that reached a different value (8 of the 34 supported rows). This error reduces the measured flawed share. Second, the letter match is case-insensitive, so an algebraic "b)" early in the text can be read as the answer statement (7 of the 32 no-derivation rows). The rollout is then labeled no derivation, which raises the measured share. Both errors apply to every setting alike.

Level-label filter control. This control asks how much of the harness effect a prior without any offline rollouts recovers. Specifically, the score of a prompt is $\ell/5$, where ℓ is its MATH difficulty label, so the selection prefers prompts with a low label. The over-draw factor is $M = 3$ and every other detail follows the harness setting of Appendix E. The control is trained with three seeds at 0.5B; its Avg8, zero-gradient rate and magnitude distribution are reported with the other 0.5B settings in Tables 4 and 14.

Prior-based advantage calibration (h1e). An earlier arm used the same offline pass to calibrate the advantage instead of selecting prompts: the positive advantages of a group were shrunk by a factor derived from the pass while the negative side was left unchanged, so the group mean is no longer zero and the estimator lies outside the family of Section 3.4. Two seeds at 0.5B reach 36.31 and 35.22 Avg8 (mean 35.77) under the clean1+clean2(aqua) protocol (Table 20). We report it for completeness; with two seeds and a different evaluation protocol it is not compared in the main text.

Model	Zero-gradient	All-wrong	All-correct	Flag rate (stale)	Coverage (stale)↑	Flag rate (refreshed)	Coverage (refreshed)↑
	share of drawn groups, GRPO			prior score > 0.5; coverage = share of zero-gradient groups whose prompt is flagged			
0.5B	36.1%	31.3%	4.8%	36.5%	72.1%	–	–
3B	48.0%	11.7%	36.3%	6.3%	9.4%	31.2%	45.3%

Table 16: Waste composition of the groups drawn by GRPO (all-wrong and all-correct shares, three-seed means), per model size. The coverage columns give the share of these zero-gradient groups whose prompt the prior flags (prior score above 0.5), for the stale and the refreshed prior.

Bucket	Before training		After 200 GRPO updates	
	correct samples	flawed↓	correct samples	flawed↓
k=1	1432	52.2%	531	49.5%
k=2–3	5626	43.9%	3842	44.6%
k=4–7	12026	40.3%	14966	41.8%
k=8–15	6929	40.8%	28956	29.5%
k=16	16	0.0%	4464	23.1%
$k\geq 8$ (pooled)	6945	40.7%	33420	28.6%

Table 17: Derivation check on correct multiple-choice rollouts by the number $k = n^+$ of correct rollouts, before training and after 200 GRPO updates (8,000 AQuA-RAT prompts, Qwen2.5-0.5B-Instruct, $G = 16$). The column "correct samples" is the count of correct rollouts and "flawed" the share of them whose derivation does not support the chosen answer or is absent.

G ALGORITHMS

Algorithm 1 lists the GRPO advantage estimator of Eq. (1), Algorithm 2 the training loop of the harness, and Algorithm 3 the reweighting estimator that serves as the reweighting-family ablation. The three construction steps of this estimator and the numerical check of the within-group ratio follow the listings.

Algorithm 1 GRPO advantage estimator (per prompt q).

Require: rollouts $\{\tau_i\}_{i=1}^G$, binary rewards $\{r_i\}_{i=1}^G$, $r_i \in \{0, 1\}$

$\mu \leftarrow \frac{1}{G}\sum_i r_i$

$\sigma \leftarrow \sqrt{\frac{1}{G}\sum_i (r_i - \mu)^2}$

for $i = 1, \ldots, G$ **do**

  $\hat{A}_i \leftarrow (r_i - \mu)/(\sigma + \varepsilon)$ ▷ per-rollout magnitude depends on (n^+, n^-)

end for

return $\{\hat{A}_i\}_{i=1}^G$

Three construction steps of the reweighting estimator. The reweighting estimator was constructed as an advantage whose magnitude does not read (n^+, n^-), in three steps that each remove one dependence of Eq. (2) on the counts. We use $y_i \in \{+, -\}$ to denote the class of rollout i, with $y_i = +$ when $r_i = 1$. First, the two classes are normalized separately,

$$\hat{A}_i^{(1)} \;=\; \frac{r_i - \mu^{y_i}}{\sigma^{y_i} + \varepsilon}, \tag{9}$$

where (μ^+, σ^+) and (μ^-, σ^-) are the mean and the standard deviation inside the correct and the wrong subgroup. A correct rollout in an imbalanced group is then no longer standardized together with many wrong rollouts. However, σ^+ still depends on n^+ and σ^- on n^-. Second, the magnitude is replaced by a global constant,

$$\hat{A}_i^{(2)} \;=\; \begin{cases} +c, & y_i = +, \\ -c, & y_i = -, \end{cases} \tag{10}$$

which removes the counts from the per-rollout magnitude. However, it breaks the zero mean of the group whenever $n^+ \neq n^-$, since the correct side sums to $n^+ c$ and the wrong side to $n^- c$. Third, the zero mean is restored with a stop-gradient rescaling of the wrong side,

$$\hat{A}^+ = c, \qquad \hat{A}^- = -\,c \cdot \mathrm{sg}\left[\tfrac{n^+}{n^-}\right], \tag{11}$$

Size	Setting	Spurious share↓	Mass rel. GRPO↓	n
0.5B	GRPO	42.3%±0.0%	1.000±0.020	3
0.5B	Level-label filter	41.7%±0.1%	1.201±0.036	3
0.5B	Shuffled (stale)	42.5%±0.1%	0.987±0.025	3
0.5B	Ours (stale)	41.8%±0.1%	1.313±0.029	3
3B	GRPO	41.5%±0.1%	1.000±0.027	3
3B	Shuffled (stale)	41.6%±0.1%	0.999±0.031	2
3B	Shuffled (refreshed)	41.6%±0.1%	1.046±0.042	6
3B	Ours (stale)	41.2%±0.1%	1.022±0.007	2
3B	Ours (refreshed)	41.4%±0.1%	1.365±0.015	6

Table 18: Expected spurious share of the positive weight and the spurious mass per drawn group relative to GRPO of the same size, for every setting on the 0.5B and 3B models. Cells report mean±SD over n seeds.

Checker label	n	judge: supported	judge: not supported	agreement (%)↑
supported	34	26	8	76
derivation mismatch	34	4	30	88
no derivation	32	7	25	78
overall agreement 81%; flawed: precision 0.83, recall 0.87; Cohen's κ = 0.59				

Table 19: Agreement between the deterministic derivation check and an LLM judge that reads the full rollout under a fixed rubric, on 100 correct rollouts sampled with a fixed seed and stratified by checker label: agreement per label, precision and recall of "flawed", and Cohen's κ.

where $\mathrm{sg}[\cdot]$ is the stop-gradient operator, so that $\sum_{y_i=+} |\hat{A}_i^{(3)}| = \sum_{y_i=-} |\hat{A}_i^{(3)}|$ while every correct rollout keeps the magnitude c. Note that the stop-gradient is notational only, since n^+/n^- is a constant of the group computed from the rewards and carries no gradient in any implementation. Eq. (11) is Algorithm 3, a replacement for Eq. (1) inside the GRPO surrogate with one hyperparameter, the scale c. In the family of Eq. (4) it is the member with $a^+ = c$. By Eq. (5) its advantage vector equals $\lambda(n^+)\,\hat{A}$ with $\lambda(n^+) = c\sqrt{n^+/n^-}$. Thus, it rescales the GRPO update of a group by a count-dependent scalar, and the direction inside the group remains the same, as stated in Section 3.4. With the trainer of Appendix E, an all-correct group is dropped before the estimator runs, so the case $n^- = 0$ in Eq. (11) does not occur.

Numerical check of the within-group ratio. Section 3.4 states that any estimator of the form of Eq. (4) has the same ratio $a^-/a^+ = -n^+/n^-$ as GRPO inside every group. We checked this statement for $G = 16$ and every $n^+ \in \{1, \ldots, 15\}$ with nine choices of a^+, among them the GRPO, Dr. GRPO, RLOO and reweighting values, each with $a^- = -a^+ n^+/n^-$ from Eq. (5). In single precision, the maximum absolute difference between $\tilde{A}/\|\tilde{A}\|$ and $\hat{A}/\|\hat{A}\|$ over the 135 cases is below 10^{-6}. Note that the check verifies the within-group ratio only, and it says nothing about the scalar weight $\lambda(n^+)$ of a group, which differs between the members of the family. The script is included with the paper sources.

H Reasoning Trace Comparisons on Bounded-Answer Benchmarks

The traces below come from the earlier comparison between GRPO and the reweighting estimator on Qwen2.5-0.5B-Instruct. They show two test problems, one from MMLU-Math and one from SAT-Math, on which the two trained policies of that study named different letters. The output labeled GRPO is listed first and the output labeled Reweighting second. In both examples the first output derives the gold value and then names a letter whose option is a different value, and the verifier does not read the derivation. We keep the traces as illustrations of answers chosen apart from the derivation on bounded-answer prompts, and we draw no comparison between methods from them.

Example 1 (MMLU-math). *Prompt.* "What is the largest factor of 54,321 that is less than 54,321? (A) 6, (B) 121, (C) 18,107, (D) 54,320." Gold answer: C.

Setting	Avg8 per seed	Avg8	Zero-grad. rate	Eval protocol
Prior-based advantage calibration (h1e)	36.31/35.22	35.77±0.77	35.1%±0.0%	clean1+clean2(aqua)
GRPO	34.17/35.22/32.93	34.11±1.14	36.1%±1.2%	full20

Table 20: The prior-based calibration arm (h1e) on the 0.5B model against GRPO. We list Avg8 per seed, its mean, the zero-gradient rate and the evaluation protocol.

Algorithm 2 Advantage Harness: over-draw and select.

Require: epoch stream $\mathcal{D}$, lookup prior $\hat{P}$ built offline (from the initial policy, or from a policy after 40 GRPO updates) and fixed for the run, over-draw factor M, prompts per update b, group size G, updates T

for $t = 1, \ldots, T$ **do**
  $\mathcal{C}_t \leftarrow$ next Mb prompts of $\mathcal{D}$ ▷ removed from the stream, none generated
  $\mathcal{S}_t \leftarrow$ the b prompts of $\mathcal{C}_t$ with the smallest $\hat{P}(q)$ ▷ Eq. (6), ties by a seeded draw
  for $q \in \mathcal{S}_t$ **do**
    sample G rollouts of q from $\pi_{\theta_{\text{old}}}$ and score them with the verifier
    if $n^+ \in \{0, G\}$ **then** drop the group ▷ zero-gradient group
    else compute $\{\hat{A}_i\}_{i=1}^{G}$ by Algorithm 1 ▷ the estimator remains the same
  end for
  update θ with the GRPO surrogate (advantage of Eq. (1)) on the kept groups ▷ bG rollouts per update, as in GRPO
end for

GRPO output (verbatim).

> To find the largest factor of 54,321 that is less than 54,321, we need to follow these steps:
>
> **1. Identify the factors of 54,321:** First, we need to find the prime factorization of 54,321. 54,321 is an odd number, so it is not divisible by 2. The sum of the digits of 54,321 is $5 + 4 + 3 + 2 + 1 = 15$, which is divisible by 3. Therefore, 54,321 is divisible by 3. Dividing 54,321 by 3, we get $54,321 \div 3 = 18,107$. Now, we need to check if 18,107 is a prime number. We do this by checking divisibility by prime numbers up to the square root of 18,107 (approximately 134.5). 18,107 is not divisible by 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, or 97. Since 18,107 is not divisible by any of these primes, it is a prime number. Therefore, the prime factorization of 54,321 is $3 \times 18,107$.
>
> **2. Identify the largest factor less than 54,321:** The largest factor of 54,321 is 18,107. To confirm, we check that 18,107 is less than 54,321. Indeed, $18,107 < 54,321$.
>
> Therefore, the largest factor of 54,321 that is less than 54,321 is $\boxed{B}$.

Reweighting output (verbatim).

> To find the largest factor of 54,321 that is less than 54,321, we need to follow these steps:
>
> **1. Identify the largest factor less than 54,321**: The largest factor of a number is its smallest positive factor. Therefore, we need to find the largest factor of 54,321 that is less than 54,321.
>
> **2. Check divisibility by small prime numbers**: We start by checking if 54,321 is divisible by the smallest prime number, 2. Since 54,321 is odd, it is not divisible by 2.
>
> **3. Check divisibility by 3**: A number is divisible by 3 if the sum of its digits is divisible by 3. The sum of the digits of 54,321 is $5 + 4 + 3 + 2 + 1 = 15$. Since 15 is divisible by 3, 54,321 is also divisible by 3. We can perform the division:
>
> $$54,321 \div 3 = 18,107$$
>
> So, $54,321 = 3 \times 18,107$.
>
> **4. Verify that 18,107 is a factor**: We need to check if 18,107 is a factor of 54,321. We do this by performing the division:
>
> $$54,321 \div 18,107 = 3$$
>
> Since the division is exact, 18,107 is indeed a factor of 54,321.
>
> **5. Conclusion**: The largest factor of 54,321 that is less than 54,321 is 18,107.
>
> Therefore, the correct answer is $\boxed{C}$.

Algorithm 3 Reweighting advantage estimator (reweighting-family ablation, per prompt q).

Require: rollouts $\{\tau_i\}_{i=1}^{G}$, binary rewards $\{r_i\}_{i=1}^{G}$, $r_i \in \{0,1\}$, global scale c

$n^+ \leftarrow |\{i : r_i = 1\}|, \quad n^- \leftarrow G - n^+$

$\rho \leftarrow \mathrm{sg}[\, n^+/n^- \,]$ ▷ a constant of the group, the stop-gradient is notational

for $i = 1, \ldots, G$ **do**

  if $r_i = 1$ **then**

    $\hat{A}_i \leftarrow c$ ▷ constant magnitude, free of (n^+, n^-)

  else

    $\hat{A}_i \leftarrow -\,c \cdot \rho$ ▷ per-class rescaling restores the zero mean

  end if

end for

return $\{\hat{A}_i\}_{i=1}^{G}$

Example 2 (SAT-Math). *Prompt.* "There are 435 voting members of the US House of Representatives. If b voting members are in favor of a certain bill, which expression represents the percentage of the voting members in favor of the bill?" Gold answer: A.

GRPO output (verbatim).

> To find the percentage of voting members in favor of the bill, we need to express the number of voting members in favor as a fraction of the total number of voting members and then convert that fraction to a percentage.
>
> The total number of voting members is 435. If b is the number of voting members in favor, then the number of voting members against is $435 - b$.
>
> The percentage of voting members in favor is given by the formula:
>
> $$\text{Percentage} = \left(\frac{b}{435}\right) \times 100$$
>
> This can be rewritten as:
>
> $$\text{Percentage} = \frac{b \times 100}{435}$$
>
> So, the expression that represents the percentage of the voting members in favor of the bill is:
>
> $$\frac{b \times 100}{435}$$
>
> Therefore, the correct answer is: $\boxed{B}$

Reweighting output (verbatim).

> To find the percentage of voting members in favor of the bill, we need to express the number of voting members in favor as a fraction of the total number of voting members and then convert that fraction to a percentage.
>
> The total number of voting members is 435. If b is the number of voting members in favor, then the number of voting members against is $435 - b$.
>
> The percentage of voting members in favor is given by the formula:
>
> $$\text{Percentage} = \left(\frac{b}{435}\right) \times 100$$
>
> This can be rewritten as:
>
> $$\text{Percentage} = \frac{b \times 100}{435}$$
>
> Therefore, the expression that represents the percentage of the voting members in favor of the bill is:
>
> $$100\left(\frac{b}{435}\right)$$
>
> So, the correct choice is:
>
> $$\boxed{A}$$

Interpretation. In both examples the first output derives the gold value and then names a letter whose option is a different value. The letter is chosen apart from the derivation, which is the behavior on bounded-answer prompts described in Section 3.2. A rollout of this kind is marked correct when the named letter happens to be the gold one and wrong otherwise. In both cases the verifier does not read the derivation. The derivation check of Section 3.3 counts the correct rollouts of this kind. Note that the traces show the behavior on two prompts and support no statement about either training method. They show the mirror case of the derivation check, a correct derivation followed by a wrong letter; the check counts correct letters without a supporting derivation and does not count this case, whose rate we do not measure.