EFFICIENT RL SELECTS THE JUDGE'S MISTAKES: PROMPT SELECTION UNDER LLM-JUDGED REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-based reinforcement learning with verifiable rewards increasingly relies on large language models as judges and data selection methods which aim to spend rollouts on mixed prompts (prompts where the group contains at least one accepted and one rejected rollout). We identify a fundamental incompatibility between these two practices: current selection methods do not distinguish between prompts the policy is genuinely learning from and prompts the judge grades incorrectly. We discover that the filtered set of prompts which the policy has not learned tends to concentrate on those which are plagued by the judge's mistakes, leading to repeated optimization toward answers the judge wrongly accepts or rejects. We show this across selection methods, model and group sizes, learning rates, and the math and science domains. With an exact verifier, these methods deliver their benefits; however, when a judge is used as the verifier, most of their advantage vanishes. In fact, up to a quarter of rollout groups train the policy toward the wrong behavior (versus 5% under uniform sampling), and the number of prompts the policy answers wrongly but the judge accepts increases two- to four-fold. As these prompts are revisited, each repeated rollout contributes less learning and more exploitation. In mathematics, held-out accuracy shows no consistent change since learned errors are largely problem specific; on the other hand, the policy uses more than double the compute it needs with an exact verifier to achieve the same mastery. On rubric-based science, judge errors appear more stylistic and exploitation transfers to out-of-distribution evaluation, costing 10 to 11 points on MMLU-Pro STEM that are not recovered within training. This loss, too, is selection amplifying the judge's (stylistic) errors, through the composition of the selected groups. We propose two simple changes to mitigate this issue: bound how much of the rollout budget any prompt may absorb, and use independent verification only on prompts that exceed this limit. Together, they use about 4% additional verifier calls and remove most of the excess exploitation. We also propose importance-weighted selection, which draws prompts in proportion to how likely the selector is to keep them and reweights each by the inverse of that probability, so the expected update equals uniform sampling's; applied to GRESO, it recovers 9 of 10 lost MMLU-Pro STEM points. Our findings call for reevaluating claims of efficiency for RL data selection under verifiers used in practice and for designing prompt selection and verification in tandem.

1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) has become the standard method to instill reasoning capabilities in large language models (LLMs) (Lambert et al., 2025; Guo et al., 2025) along with group-based on-policy gradient algorithms such as GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024). A well-known caveat is that the advantage is zero when all answers in a group are assigned the same reward, meaning compute is wasted on such zero-variance prompts. Fixes such as DAPO's dynamic sampling (Yu et al., 2025) discard them and keep sampling until enough mixed groups are collected, at a substantial cost in compute. An alternative proposed by Zheng et al. (2025a) uses the history of rewards of a prompt to predict whether it will have non-zero variance before generating rollouts, with up to 2.4 times faster rollout than dynamic sampling; similarly, Qu et al. (2026a) reported 78% fewer rollouts than dynamic sampling on Countdown. How-

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

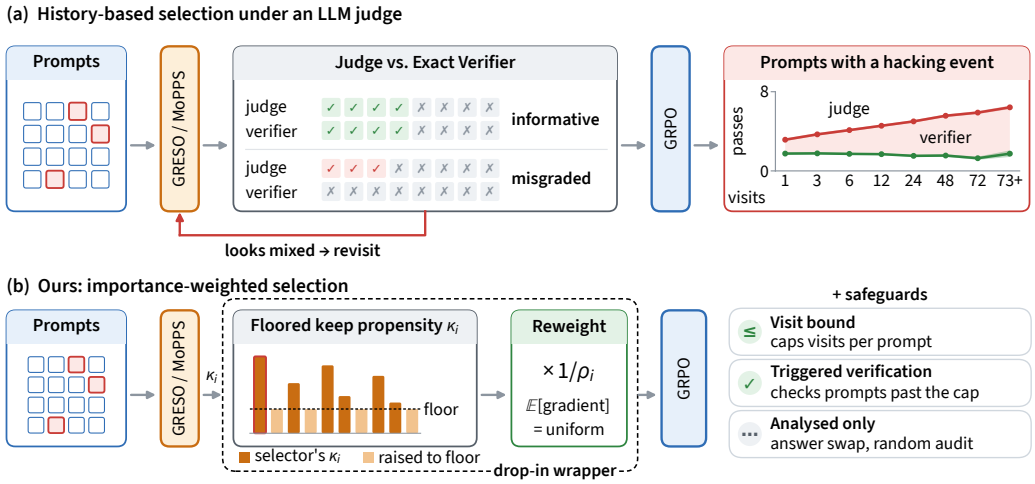


Figure 1: **(a)** A history-based selector keeps returning to prompts whose groups look mixed because the judge accepts wrong answers. The inset averages the 247 GRESO and MoPPS prompt-runs with a reward-hacking event in Math, which are partly solvable, so the verifier passes 1.3–1.8 of 8 responses while the judge's passes rise from 3.2 to 6.4 (bins labeled by upper edge; $\pm$ SE bands). **(b)** The wrapper draws prompts with probability $\rho_i \propto \kappa_i$, the selector's probability of keeping the prompt floored at 0.3, and reweights by $1/\rho_i$; we run the visit bound and triggered verification and only analyze the answer swap and random audit.

ever, these methods assume that an exact checker exists, since the reward that measures a prompt's informativeness is the same reward the policy is trained on.

Many domains, such as open-ended writing and free-response questions, lack an exact verifier, and LLMs are increasingly used as judges to assign rewards (Ma et al., 2025; Gunjal et al., 2026); these judges, like rule-based verifiers, make errors that the policy exploits (Huang et al., 2026; Mahmoud et al., 2026; Helff et al., 2026). Selecting informative instances is known to emphasize mislabeled ones (Mindermann et al., 2022; Burda et al., 2019a). Since history-based selectors were evaluated with rule-based checkers, we investigate how they behave when a fallible LLM judge assigns the reward.

We find that errors made by the judge mostly persist through training instead of being random. If a judge rewards an incorrect response to a given prompt, it will likely reward the same incorrect response when it is checked again (our Math judge re-accepts it on 68% of re-checks, against a base false-positive rate of 22%), and prompts with reward-hacking events (all eight responses wrong, at least six judged correct) recur across training runs more than difficulty-matched random prompts do. Such repeated false positives did not grow under plain GRPO with leaky code test suites (Zhang, 2026), but under history-based selection they grow with visits, since the judge's errors make groups look spuriously mixed (Figure 1a) and prompt selection methods concentrate on these groups long after they should have been skipped. Prior work has not examined this interaction, since noise corrections treat verdicts as independent of the response (Cai et al., 2025; El Mansouri et al., 2026) and data-policy comparisons used rule-based rewards (Liang et al., 2026a).

We examine the interaction between history-based selection and judge-based RLVR in Math and Science. Our policies are Qwen3-1.7B-Base and, in Science, Qwen3-8B-Base, and our judges have 12B and 26B (3.8B active) parameters; in Math, an exact grader records whether each response is correct without influencing training. There, held-out accuracy shows no consistent difference, since the learned errors are largely problem specific, but most of GRESO's lead in strict mastery (hard prompts solved in all eight responses) vanishes and the selectors need more than twice the steps to reach the same mastery. Up to a quarter of the groups GRESO trains on carry only spurious signal (5% under uniform sampling), and the selectors' reward-hacking events per visit exceed both uniform sampling's rate and chance (Section 3). In Science, GRESO and MoPPS end 10 and 11 points below uniform sampling on MMLU-Pro STEM. The rubric judge passes 82% of responses that only restate the question, as LLM judges often do for content-free responses (Zhao et al., 2025),

and selection amplifies these stylistic errors by composition, since prompts whose groups stay split are enriched for style-driven splits; the policy learns the terse style the judge rewards, which costs accuracy out of distribution. These findings point towards a trade-off that any prompt selection method relying on a single judge cannot avoid, since changing how often each prompt is visited helps under an exact checker but under a judge multiplies the worst-case weight on the judge's errors, 11.4-fold for GRESO in Math.

In this work, we propose two remedies. The first bounds how much of the rollout budget any prompt may absorb and verifies independently only the prompts past this bound; in Math it uses about 4% additional verifier calls and removes most of the excess exploitation. The second, importance-weighted selection, draws a fixed number of prompts each step with probability proportional to the selector's floored keep propensity (its probability of keeping the prompt) and weights each group by the inverse of that probability, so that the expected gradient equals uniform sampling's (Figure 1b); we reached it by treating selection as a sampling design, and it coincides up to a constant factor with DPPO's rescaling (Zhu et al., 2026a). Applied to GRESO, it recovers 9 of its 10 lost points on MMLU-Pro STEM but keeps no rollout savings of its own, and under a first-order model no unbiased design that sees only a prompt's run of consecutive zero-variance groups, as GRESO does, saves more than about a fifth of the rollouts (Theorem 1). We recommend the first remedy for keeping an existing selector, the second when a selector's larger effective sample size is wanted, and otherwise uniform sampling, or DPPO when rollouts are the binding cost.

2 PRELIMINARIES

GRPO objective. At step t a batch B_t of P prompts is drawn from a pool $\{x_i\}_{i=1}^{N}$ and the policy π_θ samples G responses y_{ij} per prompt with rewards r_{ij}. As trained (TRL's DAPO loss, no KL term; Yu et al., 2025; von Werra et al., 2020), we maximize

$$\mathcal{J}_t(\theta) = \frac{1}{Z_t} \sum_{i \in B_t} w_{t,i} \sum_{j=1}^{G} \sum_{\ell=1}^{|y_{ij}|} \min\big(\eta_{ij\ell} \hat{A}_{ij},\ \mathrm{clip}(\eta_{ij\ell}, 0.8, 1.28)\, \hat{A}_{ij}\big), \qquad \hat{A}_{ij} = \frac{r_{ij} - \bar{r}_i}{s_i + 10^{-4}}, \quad (1)$$

where $\eta_{ij\ell}$ is token ℓ's likelihood ratio to the sampling policy, $\bar{r}_i$ and s_i are the group's reward mean and sample standard deviation, $Z_t = \sum_{i,j} |y_{ij}|$, and $w_{t,i} = 1$ in unweighted arms (Appendix B.2). If a group's G rewards are equal, every $\hat{A}_{ij}$ is zero and the group adds nothing to (1); we call such prompts *zero-variance prompts* and the others *informative*. Given the history $\mathcal{F}_t$, keep propensities $\kappa_{t,i}$ (for GRESO, a function of the *zero-variance streak*, the number of consecutive zero-variance groups) induce inclusion probabilities $\rho_{t,i} = \Pr(i \in B_t \mid \mathcal{F}_t)$. The amplification (weighted exposure) $\Lambda = \max_i \sum_t \rho_{t,i} w_{t,i} / (TP/N)$ is 1 for uniform sampling ($\rho_{t,i} = P/N$); the visit amplification $\Lambda_{\mathrm{vis}} = \max_i n_{T,i} / (TP_r/N)$ counts visits $n_{t,i}$ by step t, with P_r prompts rolled out per step (Appendix B.3).

Judge error. The judge's verdict $J \in \{0, 1\}$ is the reward; in Math an exact verdict V is logged. With $a_i(y) = \Pr(J{=}1 \mid x_i, y, V{=}0)$ and $b_i(y)$ the probabilities of accepting a wrong or rejecting a correct response y, the rates $\alpha_i(\theta) = \mathbb{E}_{y \sim \pi_\theta}[a_i(y) \mid V{=}0]$ and $\beta_i(\theta)$ move with the policy, as does the judged pass rate $q_i = p_i(1-\beta_i) + (1-p_i)\alpha_i$ for true pass rate p_i. Errors are *response-dependent* when a_i or b_i varies with y, the RLVR counterpart of instance-dependent label noise (Liu et al., 2023), and *response-independent* otherwise, as assumed by noise-corrected GRPO (El Mansouri et al., 2026) and Cai et al. (2025). The Science judge passes 82% of mere restatements of the question, comparable to the rates reported for open-weight LLM judges on content-free responses (Zhao et al., 2025).

Measured quantities. A group with k^{obs} judge-accepted and k^{true} verifier-accepted responses is judge-informative if $0 < k^{\mathrm{obs}} < G$. On the hard set $\mathcal{H}$ of 359 Math prompts the base policy solves at most twice in eight attempts, visits after a judge-informative visit with $k^{\mathrm{true}} > 0$ are at risk, and a *reward-hacking event* is the first at-risk visit with $k^{\mathrm{true}} = 0$ and $k^{\mathrm{obs}} \geq 6$, reported per 1,000 at-risk visits and as *chance-adjusted events*, the count minus its chance expectation (Corollary 1). *Strict mastery* at step t is the fraction of $\mathcal{H}$ with a trained visit by step t at which $k^{\mathrm{true}} = G$. The *error-prone visit share* is the share of visits to $\hat{\mathcal{E}}$, the 106 prompts over-credited on at least half of their visits in two uniform runs. Over the last 20% of steps, the *judged-minus-true gap* δ is the mean

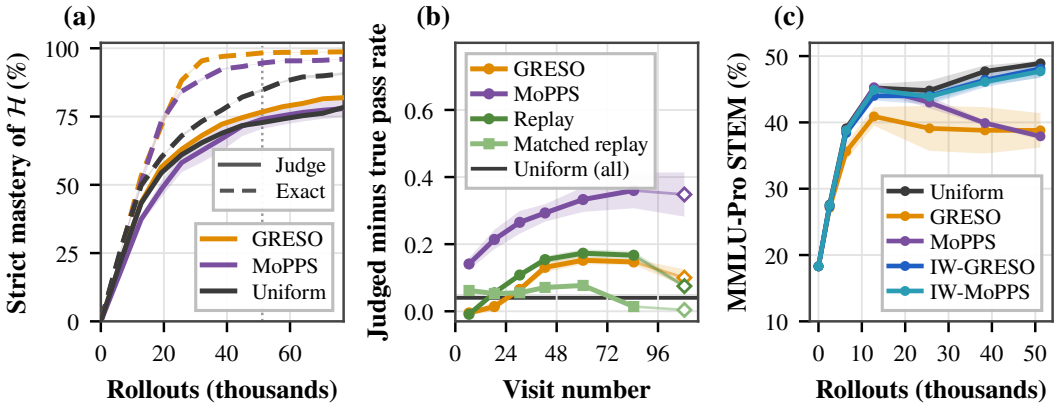


Figure 2: **(a)** Math, strict mastery of $\mathcal{H}$ with the judge (solid) or the exact verifier (dashed) as the reward; every arm rolls out $PG = 256$ responses per step, and the dotted line marks step 200. **(b)** Math, judged-minus-true gap by visit number on the prompts each run visits at least 49 times (bins of 12 visits to 48, then 24, then 97 or more, drawn as open diamonds); the uniform band is its whole-run gap over all prompts, a different cohort. **(c)** Science, MMLU-Pro STEM, with the importance-weighted arms. Shaded bands show mean ± SE over three seeds.

of $(k^{\text{obs}} - k^{\text{true}})/G$ and the *spurious-group share* the fraction of groups with $0 < k^{\text{obs}} < G$ and $k^{\text{true}} \in \{0, G\}$, called the *style-split share* in Science, where the reference judge's verdicts replace the verifier's (Appendix B.5).

3 History-based selection under an LLM judge

We train Qwen3-1.7B-Base with GRPO (Section 6). In Math, a Gemma-3-12B-it judge rewards responses to 800 MATH prompts for 300 steps, while an exact verifier grades every response without influencing training. In Science, a Gemma-4-12B-it rubric judge rewards responses to 406 RaR-Science prompts for 200 steps, and we measure transfer on MMLU-Pro STEM. Every arm has three seeds, and the Math arms are also trained with the exact verifier as the reward. We report mean ± SE over seeds; each Math selector ran twice in seeds 0 and 1, and a seed's value averages its runs (Appendix F.1).

With an exact verifier, the selectors master the hard set sooner. With the exact verifier as the reward, GRESO and MoPPS strictly master $98.3 \pm 0.1\%$ and $94.6 \pm 0.1\%$ of $\mathcal{H}$ by step 200, against $84.7 \pm 0.6\%$ for uniform sampling (Figure 2a). Held-out MATH accuracy is unchanged, consistent with the savings reported by Zheng et al. (2025a) and Qu et al. (2026a). Since strict mastery is recorded only on trained visits, part of this lead reflects revisits (Appendix B.5).

Judge errors depend on the response and recur across runs. The Math judge re-accepts a wrong response it has accepted on $68 \pm 5\%$ of re-checks (response-cluster bootstrap SE), against 49% for another wrong response to the same prompt and a base false-positive rate of 22% (Appendix C). The prompts with a reward-hacking event recur, with a Jaccard overlap of 0.20 to 0.47 between selector runs and 0.21 to 0.28 between uniform runs. Random sets from the same difficulty strata overlap at only 0.09 to 0.13 (Appendix F.4), so the recurrence points to the judge's errors on particular prompts.

Under the judge, the selectors concentrate their updates on the judge's errors. Most of the advantage in mastery vanishes under the judge. GRESO's lead falls from 13.6 to 3.9 ± 0.7 points, and MoPPS's ranges from -18 to $+8$ across runs (Figure 2a). Reaching strict mastery of 80% of $\mathcal{H}$ takes GRESO 2.4 to 3.1 times the steps of the same seed's exact-verifier run and MoPPS 2.2 to 2.9 times. Three runs never reach it (Appendix F.5), so the policy uses more than double the compute for the same mastery. In the last fifth of training $20.7 \pm 0.9\%$ and $19.2 \pm 1.2\%$ of the selectors' groups are spurious, up to about a quarter in single runs, against $5.2 \pm 0.1\%$ under

Table 1: Replay control (Math). "Source runs" are the GRESO runs whose schedules are replayed (one per seed), so they differ from GRESO's arm mean in Table 2b. Both replays have $\Lambda_{\text{vis}} = 11.47 \pm 0.05$. Error-prone visits land on $\hat{\mathcal{E}}$, the 106 prompts over-credited on at least half of their visits in two uniform runs; events beyond chance subtract Corollary 1's chance count; mastery is at step 200. Mean ± SE over three seeds; bold marks the best value in each outcome column.

	Error-prone visits (%) ↓	Events beyond chance ↓	Events per 1k $\mathcal{H}$ visits ↓	Gap δ ↓	Spurious groups (%) ↓	Strict mastery (%)
Source runs	28.4 ± 0.1	12.6 ± 0.9	4.88 ± 0.43	0.123 ± 0.028	21.1 ± 2.4	76.9 ± 0.3
Replay	28.4 ± 0.1	10.2 ± 0.8	4.11 ± 0.25	0.117 ± 0.007	24.3 ± 0.6	75.2 ± 0.3
Matched replay	16.1 ± 0.2	$\mathbf{0.5 \pm 0.1}$	$\mathbf{1.69 \pm 0.16}$	0.043 ± 0.002	5.4 ± 0.2	67.5 ± 0.4
Uniform	13.3	0.6 ± 0.1	2.63 ± 0.08	$\mathbf{0.040 \pm 0.002}$	$\mathbf{5.2 \pm 0.1}$	72.8 ± 0.3
Replay − match	$+12.3 \pm 0.2$	$+9.7 \pm 0.8$	$+2.41 \pm 0.11$	$+0.074 \pm 0.005$	$+18.9 \pm 0.6$	$+7.8 \pm 0.6$

uniform sampling. The selectors drive two to four times as many prompts of $\mathcal{H}$ into reward-hacking events per run, and after subtracting the events that response-independent errors would produce by chance (Corollary 1), GRESO and MoPPS keep 10.8 ± 0.6 and 18.5 ± 0.7 events against 0.6 ± 0.1 for uniform sampling. The selectors create about three times as many at-risk visits and produce events on them at uniform sampling's rate, although chance expects a lower rate on their visits, so both exposure and response-dependent error contribute (Appendix F.1). Repeated visits thus bring less learning and more exploitation, as 93% of the prompts uniform sampling masters are mastered within six visits, while 69% to 87% of the selectors' events occur later. The gap on heavily revisited prompts grows over the first 48 visits (Figure 2b), consistent with Proposition 3, whereas Zhang (2026) found no such growth under plain GRPO. Held-out MATH accuracy shows no consistent change, in line with learned errors that are largely specific to their problems.

At equal amplification, the harm follows the repeated prompts. A same-seed GRESO run's recorded prompt schedule, replayed into a fresh policy whose rewards cannot influence which prompts it trains on, reproduces the harm, while the same schedule with prompts permuted among those of equal base pass count does not (Table 1). Per 1,000 visits to $\mathcal{H}$ the replay produces 4.1 events, against 1.7 for the matched replay and 2.6 for uniform sampling. The policy's own feedback is therefore not needed, and since the matched replay also masters fewer prompts, we read the contrast as showing that the identity of the repeated prompts matters.

In Science, selection enriches batches for style-split groups, and the policy learns a costly style. By the end of training, GRESO and MoPPS are 10.1 ± 2.1 and 11.0 ± 1.0 points below uniform sampling's $48.9 \pm 0.4\%$ on MMLU-Pro STEM, a loss they do not recover (Figure 2c). GRESO falls behind within 25 steps and then stays flat, while MoPPS matches uniform sampling to step 50 and then declines steadily. Under the weak rubric judge much of the within-group variance is stylistic, since the judge passes 82% of responses that merely restate the question, comparable to the rates prior work reports for open-weight LLM judges on content-free responses (Zhao et al., 2025). A correctness-driven split resolves as the policy learns while a style-driven one persists, and under uniform sampling $31.7 \pm 1.0\%$ of the groups that follow two or more consecutive informative visits are style-split, against $15.1 \pm 0.3\%$ on a first informative visit. History-based selectors keep exactly these persistently split prompts, so from the first steps their batches carry more style-split groups than those of uniform or dynamic sampling, which selects on the current step's split rather than on history (Figure 3c). This is a composition effect, which is why GRESO falls behind while its visit amplification is still no larger than dynamic sampling's; MoPPS's share keeps rising, and so does its loss. The policy then learns the style the judge rewards. By step 200 about half of the selectors' MMLU-Pro answers are terse, against 11% for uniform sampling, and their training responses restate rubric terms twice as often (Appendix F.1). The terse style costs accuracy where derivation is needed, since the loss is concentrated on the items that require a multi-step derivation (16 to 17 points, against 2 to 3 on the rest). Consistent with this account, the loss shrinks with the stronger Gemma-4-26B-A4B-it judge, which splits fewer groups on style, is not detectable with the reference judge as the reward, and is mostly recovered by importance-weighted selection, which restores uniform sampling's share of style-split groups (Section 6), while held-out rubric pass under

the stronger judge stays similar. The effect is largest for weak judges and persists at a group size of 16, at other learning rates and with Qwen3-8B-Base (Appendices F.15, F.16 and F.13).

4 WHY IT HAPPENS

These results hold for any judge that grades each response on its own (proofs in Appendix D).

Proposition 1 (Judge errors keep prompts looking informative)**.** *A group is informative with probability $1-q_i^G-(1-q_i)^G$; for $G=8$, a prompt the policy never solves is informative with probability 0.832 at a false-positive rate of 0.2.*

Proposition 2 (Amplification bounds the first-order bias)**.** *If judge errors bias the gradients of at most m prompts by at most Δ, a selector's cumulative first-order bias is at most Λ times uniform sampling's worst case.*

Unsolvable prompts thus keep looking informative and draw the selector back; amplification caps the exposure, but which prompts are repeated decides the harm (Section 3).

Proposition 3 (Response-independent error only rescales the true update; response-dependent error raises the gap)**.** *The expected group gradient is $\omega_G(q_i)\nabla q_i$ with $\omega_G > 0$ (Lemma 1, after Davis & Recht, 2025; Mroueh, 2025), where $\nabla q_i = (1-\alpha_i-\beta_i)\nabla p_i + C_i$ and $C_i = (1-p_i)\nabla\alpha_i - p_i\nabla\beta_i$. If errors do not depend on the response, $C_i = 0$ and a prompt's own updates cannot raise its judged-minus-true gap; otherwise each of them raises it by about $\eta\,\omega_G(q_i)\,C_i^\top H C_i$ on a prompt the policy rarely solves.*

C_i, the covariance term of Cai et al. (2025, App. B), is the policy's ability to raise its own false-positive rate, so response-dependent error grows with visits (Figure 2b). Splits driven by style rather than correctness do not resolve with learning, so selectors enrich their batches for them and the policy learns the style (Appendix D.2).

Corollary 1 (Reward hacking requires response-dependent error)**.** *Under response-independent error α_i does not change with training, so beyond chance a reward-hacking event needs response-dependent error, interference from other prompts or noise in the prompt's own updates; a visit with eight wrong responses yields an event by chance with probability $\Pr(\mathrm{Bin}(8,\alpha_i)\geq 6)$.*

Summed over at-risk visits with each prompt's earlier rate, this gives the chance expectation that Section 3 subtracts; the excess is 0.6 ± 0.1 events under uniform sampling against 10.8 ± 0.6 for GRESO (Table 14).

Observation 1 (What one judge cannot detect)**.** *A wrong response accepted with probability c and a correct one rejected with probability $1-c$ yield the same transcripts and verdicts, however often the judge is re-asked, so detecting such errors needs an outside reference.*

5 BOUNDED, VERIFIED VISITS AND UNBIASED SELECTION

5.1 THE VISIT BOUND WITH TRIGGERED VERIFICATION

The first change skips a prompt while its visits exceed $\Lambda_{\max} = 1.5$ times its uniform share tP/N; the second verifies only the prompts that exceed this limit. Combined, a prompt at the bound keeps being drawn, an independent verifier (the exact verifier in Math) checks the responses the judge accepts, and the prompt is retired after two consecutive rejections or exempted until its visits double (Algorithm 2, Appendix D.10). The bound caps each prompt's exposure and hence the first-order bias of the judge's errors (Proposition 2), and verification is spent where response-dependent error concentrates (Proposition 3). Applied to GRESO in Math, the bound alone removes most of the excess reward-hacking events, and with verification the judged-minus-true gap falls to 0.012 ± 0.001, below uniform sampling's 0.040 ± 0.002. This costs about 4% additional verifier calls (416 verified visits over 9,600 trained groups). We also analyze an answer-swap reward, which fails against a judge that checks whether the reasoning supports the answer, and a random audit by a reference judge, which must grade about 65% of the groups for the rubric judge (Appendices D.11 and D.12).

5.2 IMPORTANCE-WEIGHTED SELECTION

Proposition 4 (Unbiased selection)**.** *Let the wrapper floor a selector's keep propensity $\kappa_{t,i}$ at $\kappa_{\min} = 0.3$, set $\rho_{t,i} \propto \kappa_{t,i}$ with $\sum_i \rho_{t,i} = P$, draw exactly P prompts by systematic probability-proportional-to-size sampling, scale each drawn group's advantages by $w_{t,i} = P/(N\rho_{t,i})$ and fix the loss normalizer before the draw (Algorithm 1). Then the expected gradient given $\mathcal{F}_t$ equals uniform sampling's, and every weighted exposure is P/N, so $\Lambda = 1$.*

The weighted update is the Horvitz–Thompson estimator (Horvitz & Thompson, 1952), also used by InfoBatch (Qin et al., 2024); we arrived at it by treating selection as a sampling design, and it coincides up to a constant factor with DPPO's prompt-level rescaling (Zhu et al., 2026a) (conditions in Appendix D.9).

Theorem 1 (Savings bound)**.** *Let $M_{t,i}$ be the second moment of prompt i's group gradient given $\mathcal{F}_t$. An unbiased design needs at least $v_t^\star = (N^{-1}\sum_i \sqrt{M_{t,i}})^2/(N^{-1}\sum_i M_{t,i})$ times uniform sampling's rollouts to match the design-dependent part of its second moment, with equality for $\rho_{t,i} \propto \sqrt{M_{t,i}}$.*

This is the square-root rule of importance-sampling SGD (Zhao & Zhang, 2015) (Appendix D.8), and its savings come from prompts that are never informative. Under a first-order model evaluated on the uniform runs, designs whose propensity depends only on GRESO's zero-variance streak need at least 79.1% of uniform sampling's rollouts in Math after step 50 and 94.8% in Science; GRESO's further savings come from its change of objective. The wrapper keeps no rollout savings, since it draws P prompts every step; per rollout, IW-GRESO has 7% more verifier-informative effective groups than uniform sampling in Math and 5% fewer judge-informative ones in Science. DPPO reaches similar safety at 0.6 of the rollouts, paying with a design term at least $v_t^\star/0.6$ times uniform sampling's; unlike DPPO, the wrapper uses an existing selector's propensities, draws from the whole pool instead of pruning uniformly drawn batches, fixes the token normalizer before the draw and caps every weight at $1/\kappa_{\min}$. Section 6 separates the floor from the weights. The bound with triggered verification is thus the cheap remedy that keeps any selector, importance-weighted selection the principled control for a selector already in use, uniform sampling preferable where few prompts are never informative (Science), and DPPO where rollouts dominate the cost.

6 EXPERIMENTS

Setup. We train Qwen3-1.7B-Base (Yang et al., 2025) with GRPO ($P = 32$, $G = 8$). In Math, Gemma-3-12B-it (Gemma Team et al., 2025) rewards responses to 800 MATH prompts (Hendrycks et al., 2021) without the reference answer while an exact verifier grades them, and we evaluate on 500 held-out problems. In Science, Gemma-4-12B-it (Gemma Team et al., 2026) grades responses to 406 RaR-Science prompts (Gunjal et al., 2026) against rubrics, and we evaluate on MMLU-Pro STEM, the physics, chemistry and biology questions of MMLU-Pro (Wang et al., 2024). We also run DPPO (Zhu et al., 2026a) and Qwen3-8B-Base (Appendix E). Every run has three seeds; we report mean $\pm$ SE, with differences from uniform sampling paired within a seed. Analysis plans preceded each run (Appendix J); main-text runs used about 5,500 A100 GPU-hours, about 10,200 in all (Appendix E.3).

Science. GRESO and MoPPS end 10.1 ± 2.1 and 11.0 ± 1.0 points below uniform sampling on MMLU-Pro STEM, while dynamic sampling (DS) matches it (Table 2a). Importance weighting recovers 9 of GRESO's 10 lost points, while the floored draw without weights loses 4.6 ± 0.9 (Appendix F.8), since the weights restore uniform sampling's expected composition of groups, which the mechanism of Section 3 requires. The loss is largest for weak rubric judges and is under 1 point with a reference-based judge (Table 2c).

Math. Held-out MATH accuracy shows no consistent difference (Table 2b), but GRESO's lead in strict mastery falls from 13.6 ± 0.4 points with the exact verifier as reward to 3.9 ± 0.7 under the judge, and its events exceed chance by 10.8 ± 0.6 (Figure 3a). Bounding each prompt at 1.5 times uniform sampling's share of visits halves GRESO's events. Verifying the prompts above the bound and retiring those the judge accepts wrongly costs 4.3% more verifier calls; together the two changes bring the excess over chance to 1.0 ± 0.2 and the gap to 0.012 ± 0.001. Importance weighting needs

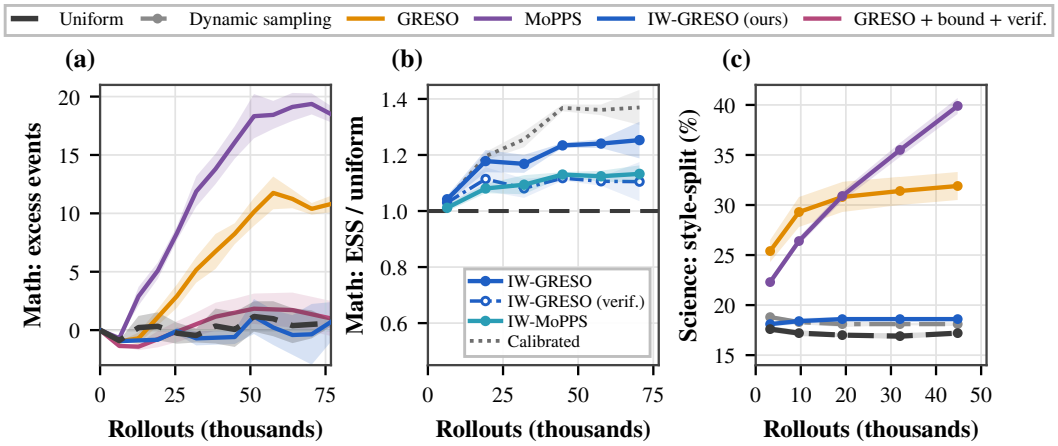


Figure 3: **(a)** Math chance-adjusted events on $\mathcal{H}$ over training: both mitigations stay at uniform sampling's level. **(b)** Math Kish effective sample size relative to uniform sampling per 50-step window (the calibrated rule of Theorem 1 is a counterfactual evaluated on the uniform runs' records; GRESO and MoPPS, at about 2.2, are omitted). **(c)** Science share of style-split groups among trained groups (Section 3). Shaded bands show mean $\pm$ SE over three seeds.

no verifier and reaches uniform sampling's level, at a cost of 2.5 ± 0.2 points of strict mastery against GRESO. It keeps no rollout savings; its effective sample size exceeds uniform sampling's by 16% over judge-informative and 7% over verifier-informative groups (Figure 3b).

Scope and choice of method. With a 3,200-prompt pool, where uniform sampling visits each prompt about three times, GRESO's excess over chance shrinks to 6.1 ± 0.4 but persists; with Qwen3-8B-Base in Math it is 7.7 ± 0.8. Both mitigations remove it in each case (Appendices F.10 and F.11), and the effects persist across group sizes and learning rates (Appendices F.15 and F.16); Section 5 discusses which method to choose.

7 Related Work

Prompt selection and imperfect judges. History-based selectors skip or replay prompts by their past rewards (Zheng et al., 2025a; Qu et al., 2026a; Zhu et al., 2026a;b; Liang et al., 2026b; Baroian & Berger, 2026) and, like a comparison of thirteen data policies (Liang et al., 2026a), used rule-based checkers. Noise corrections assume response-independent verdicts (El Mansouri et al., 2026) or identify the covariance dependence adds (Cai et al., 2025); response-dependent false positives do not grow under plain GRPO (Zhang, 2026), but under history-based selection they do. LLM judges also reward surface features such as length and content-free openers (Zheng et al., 2023; Zhao et al., 2025), which policies learn (Singhal et al., 2024; Mahmoud et al., 2026); selection enriches training for the groups such features split.

Bounded reuse and targeted verification. Prompt Replay caps reuse (Baroian & Berger, 2026) and LEEPS warns that reuse narrows coverage (Liang et al., 2026b), both against overfitting under exact rewards, and RLAVR buys ground-truth labels for chosen samples (Wang et al., 2026b). Our bound decides which prompts are verified, since the prompts that exceed it carry the judge's errors.

Importance-weighted selection. As in importance-weighted active learning (Beygelzimer et al., 2009), inverse-probability weights remove selection bias in InfoBatch (Qin et al., 2024) and DPPO (Zhu et al., 2026a), whose rescaling ours matches up to a constant factor. Ours wraps any selector but keeps no rollout savings; DPPO reaches similar safety with 0.6 of the rollouts (Section 6; Appendix A).

Table 2: Final results, mean ± SE over three seeds (per seed in Appendix F.7, more baselines in Table 40). Δ is the paired difference from uniform sampling in points, * marks weighted rows ($\Lambda = 1$), rollouts and cost are relative to uniform sampling, and strict mastery, which favors revisiting, is not bolded.

(a) Science (Gemma-4-12B-it rubric judge, 200 steps).

		Qwen3-1.7B-Base			Qwen3-8B-Base	
Method	Rollouts ↓	STEM (%) ↑	Δ STEM	Λ_{vis} ↓	STEM (%) ↑	Δ STEM
Uniform	1.00	48.9 ± 0.4	–	1.08	68.6 ± 0.5	–
DS	1.50	$\mathbf{49.0 \pm 0.3}$	$+0.1 \pm 0.7$	**1.06**	$\mathbf{68.8 \pm 0.4}$	$+0.2 \pm 0.6$
GRESO	1.00	38.8 ± 2.6	-10.1 ± 2.1	5.96 ± 0.22	61.7 ± 1.1	-6.9 ± 0.6
MoPPS	1.00	37.9 ± 0.5	-11.0 ± 1.0	9.16 ± 0.22	60.5 ± 1.1	-8.2 ± 0.6
DPPO*	**0.59**	47.0 ± 1.0	-1.9 ± 0.7	1.08	67.3 ± 1.1	-1.4 ± 0.7
IW-GRESO*	1.00	48.1 ± 1.3	-0.8 ± 1.0	2.20 ± 0.04	68.0 ± 1.1	-0.6 ± 0.7
IW-MoPPS*	1.00	47.7 ± 1.0	-1.2 ± 0.7	2.71 ± 0.08	67.7 ± 1.0	-0.9 ± 0.6

(b) Math (Gemma-3-12B-it, 300 steps; mastery of $\mathcal{H}$ at step 200; chance-adjusted = events minus chance).

		MATH	Strict mastery (%), reward:		Events	Chance-
Method	Cost ↓	held-out (%) ↑	verifier	judge	on $\mathcal{H}$ ↓	adjusted ↓
Uniform	1.00	82.4 ± 0.8	84.7 ± 0.6	72.8 ± 0.3	$\mathbf{11.3 \pm 0.3}$	$\mathbf{+0.6 \pm 0.1}$
DS	1.50	$\mathbf{82.9 \pm 0.4}$	86.8 ± 1.2	74.5 ± 0.4	12.0 ± 1.5	$\mathbf{+0.6 \pm 0.2}$
GRESO	1.00	81.5 ± 1.3	98.3 ± 0.1	76.7 ± 0.5	29.8 ± 0.2	$+10.8 \pm 0.6$
MoPPS	1.00	81.9 ± 1.6	94.6 ± 0.1	74.0 ± 3.5	32.3 ± 0.7	$+18.5 \pm 0.7$
DPPO*	**0.60**	82.2 ± 0.4	85.1 ± 1.3	72.5 ± 0.3	12.0 ± 1.2	$+0.7 \pm 0.1$
IW-GRESO*	1.00	82.8 ± 0.3	87.3 ± 1.2	74.2 ± 0.2	12.0 ± 1.7	$+0.7 \pm 0.2$
IW-MoPPS*	1.00	82.7 ± 0.2	86.6 ± 1.2	73.8 ± 0.2	13.0 ± 1.7	$+0.9 \pm 0.1$
GRESO, bound	1.00	$\mathbf{82.9 \pm 0.2}$	–	76.7 ± 0.2	15.0 ± 1.7	$+2.4 \pm 0.3$
+ verification	1.00 + 4.3%	82.2 ± 0.4	–	79.4 ± 0.6	17.3 ± 1.5	$+1.0 \pm 0.2$

(c) Judge strength (Science, 1.7B). Agreement: binary pass agreement with GLM-5.3-Flash (GLM-5 Team et al., 2026) and Kimi-K2.6 (Kimi Team et al., 2025) (binomial SE).

Training judge	Agreement ↑	Uniform STEM (%)	Δ GRESO	Δ MoPPS
Gemma-4-12B-it	0.764 ± 0.011	48.9 ± 0.4	-10.1 ± 2.1	-11.0 ± 1.0
Gemma-4-26B-A4B-it	0.941 ± 0.006	50.2 ± 0.3	-3.4 ± 0.9	-2.7 ± 0.8
DeepSeek-V4-Flash, with reference	$\mathbf{0.964 \pm 0.005}$	50.8 ± 0.4	-0.9 ± 1.0	-0.5 ± 0.6

8 Limitations and conclusion

Limitations and scope. Our evidence covers two domains, Qwen3 policies of 1.7B and 8B parameters, 200 to 300 steps, rubric judges of 12B and 26B (3.8B active) parameters with a reference-answer judge, and three seeds per arm; Qwen-family policies gain even from spurious rewards (Shao et al., 2025), so accuracy levels may not transfer. The Science effect is largest under the weaker rubric judge, which passes 82% of responses that only restate the question, comparable to the false-positive rates reported for open-weight LLM judges on content-free responses (Zhao et al., 2025); the Science ground truth is itself a panel of two language models. The savings bound of Theorem 1 holds under a first-order model, and the answer swap and random audit are analyzed rather than run.

Conclusion. Under an LLM judge, history-based selection keeps returning to prompts the judge grades wrongly. In Math it produces reward-hacking events well beyond chance; in Science it enriches batches for groups the judge splits on style, a style the policy learns and carries out of distribution at a cost of 10 to 11 points on MMLU-Pro STEM. A visit bound with verification of only the prompts that exceed it removes most of the excess exploitation for about 4% additional verifier calls, and importance-weighted selection recovers 9 of GRESO's 10 lost points but keeps no rollout savings of its own. Efficiency claims for RL data selection should be reevaluated under the verifiers used in practice, with prompt selection and verification designed together.

AI USE STATEMENT

The research ideas, the design of the study and the initial derivations of the mathematical results were developed entirely by the authors. We used large language models to write the code that executes the experiments, to generate figures, and to polish and aid the writing of the paper. We reviewed all AI-assisted work, and we take responsibility for the final content of this work, including its text, claims and code.

ETHICS STATEMENT

This work involves no human subjects, no personal data and no data collected from people; all prompts come from public datasets used under their licenses (Appendix I), and the reference labels for Science come from language models rather than annotators. Our subject is a failure mode of training with learned rewards, in which a policy learns responses that a judge accepts although they are wrong. Documenting it, and showing that a common family of efficiency methods makes it worse, is intended to help practitioners detect and limit reward hacking; the examples in Appendix H illustrate exploits of the kind already documented for LLM judges (Mahmoud et al., 2026; Zhou, 2026). The trained policies are small research checkpoints that we do not deploy. The reported runs used about 10,200 A100 GPU-hours, plus about 300 GPU-hours of development runs, and about 0.7 million reward calls to a hosted judge (Appendix E.3).

REPRODUCIBILITY STATEMENT

Each training run in the paper was run with three seeds (the Math selectors with two runs at seeds 0 and 1, averaged within the seed; Appendix J), and the uncertainty of a training-run quantity is a standard error over the three seed means; judge error rates carry binomial or bootstrap standard errors. Section 2 and Appendix B define the metrics; Appendix C gives the judge prompts, rubric templates, the re-check protocol and the calibration against the reference panel; Appendix D contains the proofs, the pseudocode of the importance-weighted selection (Algorithm 1) and the estimators; Appendix E lists hyperparameters, baseline implementations and the compute used by each group of runs; Appendix J lists the analysis plans written before the runs and their amendments in order. We will release the training and analysis code, the selector implementations, the prompt indices of all pools and held-out sets, and the per-group judge and verifier verdict logs of all runs (Appendix K); an anonymous copy of the code is provided as supplementary material.

REFERENCES

Johannes Ackermann, Michael Noukhovitch, Takashi Ishida, and Masashi Sugiyama. Gradient regularization mitigates reward hacking in reinforcement learning from human feedback and verifiable rewards. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.18037.

Fariya Afrin and Ibne Farabi Shihab. Evaluator ensembles under reward hacking: Covariance geometry and finite-search guarantees. *arXiv preprint arXiv:2608.08002*, 2026.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2402.14740.

Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance reduction in SGD by distributed importance sampling. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2016. URL https://arxiv.org/abs/1511.06481.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026. URL https://arxiv.org/abs/2504.03380.

Andrei Baroian and Rutger Berger. Prompt replay: Speeding up GRPO with on-policy reuse of high-signal prompts. *arXiv preprint arXiv:2603.21177*, 2026.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *International Conference on Machine Learning (ICML)*, pp. 49–56, 2009. URL https://arxiv.org/abs/0812.4952.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations (ICLR)*, 2019a. URL https://arxiv.org/abs/1808.04355.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019b. URL https://arxiv.org/abs/1810.12894.

Xin-Qiang Cai, Wei Wang, Feng Liu, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Reinforcement learning with verifiable yet noisy rewards under imperfect verifiers. *arXiv preprint arXiv:2510.00915*, 2025.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1704.07433.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1905.00360.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. ODIN: Disentangled reward mitigates hacking in RLHF. In *International Conference on Machine Learning (ICML)*, pp. 7935–7952, 2024. URL https://arxiv.org/abs/2402.07319.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for LLM reasoning. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2505.14970.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 23, pp. 442–450, 2010.

Dominik Csiba and Peter Richtárik. Importance sampling for minibatches. *Journal of Machine Learning Research*, 19(27):1–21, 2018. URL https://arxiv.org/abs/1602.02283.

Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2404.04475.

Kazuki Egashira, Mark Vero, Jasper Dekoninck, Florian E. Dorner, Robin Staab, and Martin Vechev. Delay, plateau, or collapse: Evaluating the impact of systematic verification error on RLVR. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2605.02909.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? Reward model ensembles mitigate but do not eliminate reward hacking. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2312.09244.

Omar El Mansouri, Fathinah Asma Izzati, Mohamed El Amine Seddik, and Salem Lahlou. Noise-corrected GRPO: From noisy rewards to unbiased gradients. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2510.18924.

Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. Reinforcement learning with a corrupted reward channel. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4705–4713, 2017. URL https://arxiv.org/abs/1705.08417.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. LILO: Learning to reason at the frontier of learnability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.12272.

Chengguang Gan, Zhixi Cai, Yunhao Liang, Hanjun Wei, Shiwen Ni, and Qinghao Zhang. A learning-rate-gated failure of GRPO in a small language and vision-language model web agent: A controlled null and its mechanism. *arXiv preprint arXiv:2607.12640*, 2026.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, pp. 10835–10866, 2023. URL https://arxiv.org/abs/2210.10760.

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 (ICLR)*, 2026. URL https://arxiv.org/abs/2510.01135.

Gemma Team, Aishwarya Kamath, Johan Ferret, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Niloofar Gholipour, Marcos Assuncao, Gursimran Singh, Timothy Yu, Rajkumar Buyya, Julien Gascon-Samson, Zhenan Fan, Yong Zhang, Xiaojie Xu, Yaqiang Yao, and Xiaolong Bai. Rollout efficiency in reinforcement learning for reasoning large language models: A taxonomy and future directions. *arXiv preprint arXiv:2609.25463*, 2026.

GLM-5 Team, Aohan Zeng, et al. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

V. P. Godambe and V. M. Joshi. Admissibility and Bayes estimation in sampling finite populations. I. *The Annals of Mathematical Statistics*, 36(6):1707–1722, 1965.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2507.17746.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://arxiv.org/abs/2501.12948.

Chaoqun He, Renjie Luo, Yuzhuo Bai, et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2402.14008.

Jujie He, Jiacai Liu, Chris Yuhao Liu, et al. Skywork Open Reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Lukas Helff, Quentin Delfosse, David Steinmann, Ruben Härle, Hikaru Shindo, Patrick Schramowski, Wolfgang Stammer, Kristian Kersting, and Felix Friedrich. LLMs gaming verifiers: RLVR can lead to reward hacking. *arXiv preprint arXiv:2604.15149*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Daniel G. Horvitz and Donovan J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

Zhibo Hou, Zhiyu An, and Wan Du. Beyond Noisy-TVs: Noise-robust exploration via learning progress monitoring. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.25438.

Xuchun Hu. CREDO: Variance-guided rubric evolution for replay-corrected credit assignment. *arXiv preprint arXiv:2609.24174*, 2026.

Yuzhen Huang, Weihao Zeng, Xingshan Zeng, Qi Zhu, and Junxian He. From accuracy to robustness: A study of rule- and model-based verifiers in mathematical reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. URL https://arxiv.org/abs/2505.22203.

Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. VCRL: Variance-based curriculum reinforcement learning for large language models. *arXiv preprint arXiv:2509.19803*, 2025.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning (ICML)*, pp. 4940–4950, 2021. URL https://arxiv.org/abs/2010.03934.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1803.00942.

Zachary Kenton, Lili Janzer, Rory Greig, Tian Huey Teh, Kirill Tyshchuk, Jonah Brown-Cohen, Harri Edwards, Senthooran Rajamanoharan, Noah Y. Siegel, Natasha Jaques, and Rohin Shah. Debate training reduces reward hacking in RLAIF. *arXiv preprint arXiv:2608.17776*, 2026.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.13786.

Yujin Kim, Namgyu Ho, Sangmin Hwang, Joonkee Kim, Yongjin Yang, Sangmin Bae, Seungone Kim, Jaehun Jung, Se-Young Yun, and Hwanjun Song. LLM-as-a-Tutor: Policy-aware prompt adaptation for non-verifiable RL. *arXiv preprint arXiv:2607.04412*, 2026.

Kimi Team, Yifan Bai, Yiping Bao, et al. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Leslie Kish. *Survey Sampling*. John Wiley & Sons, New York, 1965.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2411.15124.

Lucien Le Cam. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, 1(1):38–53, 1973.

Aitor Lewkowycz, Anders Andreassen, David Dohan, et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.14858.

Hao Liang, Mingrui Chen, Hengyi Feng, Meiyi Qiang, and Wentao Zhang. DataFlex-RL: An evaluation platform for RLVR data policies. *arXiv preprint arXiv:2609.06107*, 2026a.

Shuang Liang, Haoyang Zhou, Yifan Gong, Guowei Wang, and Xiting Wang. LEEPS: Latent-guided explore-exploit prompt sampling for efficient RLVR in large language models. *arXiv preprint arXiv:2607.28077*, 2026b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.20050.

Yang Liu, Hao Cheng, and Kun Zhang. Identifiability of label noise transition matrix. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2202.02016.

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. In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2503.20783.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. SCP-116K: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *arXiv preprint arXiv:2501.15587*, 2025.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-Reasoner: Advancing LLM reasoning across all domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.14652.

William G. Madow. On the theory of systematic sampling, II. *The Annals of Mathematical Statistics*, 20(3):333–354, 1949. doi: 10.1214/aoms/1177729988.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2605.12474*, 2026.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2206.07137.

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

Rémi Munos. Error bounds for approximate policy iteration. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 560–567, 2003.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. URL https://arxiv.org/abs/1310.5715.

Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2602.01601.

Michael Noukhovitch, Hamish Ivison, Nathan Lambert, and Aaron Courville. Learning to solve hard problems in RL for LLMs by never giving up. *arXiv preprint arXiv:2609.13443*, 2026.

Pierre-Yves Oudeyer, Frédéric Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024. URL https://arxiv.org/abs/2407.06551.

Andreas Plesner, Francisco Guzmán, and Anish Athalye. An imperfect verifier is good enough: Learning with noisy rewards. *arXiv preprint arXiv:2604.07666*, 2026.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. InfoBatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2303.04947.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating RL finetuning of reasoning models? In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2026a. doi: 10.1145/3770854.3780263. URL https://arxiv.org/abs/2507.04632.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, and Xiangyang Ji. Small generalizable prompt predictive models can steer efficient RL post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026b.

Ali Rad, Khashayar Filom, Darioush Keivan, Peyman Mohajerin Esfahani, and Ehsan Kamalinejad. Rate or Fate? RLV$^{\varepsilon}$R: Reinforcement learning with verifiable noisy rewards. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2601.04411.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-Proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2311.12022.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, and Zhipeng Wang. Not all tokens are needed (NAT): Token efficient reinforcement learning. *arXiv preprint arXiv:2603.06619*, 2026.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.05952.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pp. 489–511, 2013.

Tommy Sha, Skylar Zhai, and Siqi Zhao. ThinkPrior: Zero-rollout difficulty priors for cold-start prompt selection in RLVR. *arXiv preprint arXiv:2609.09075*, 2026.

Rulin Shao, Shuyue Stella Li, Rui Xin, 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, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wei Shen, Guanlin Liu, Zheng Wu, Ruofei Zhu, Qingping Yang, Chao Xin, Yu Yue, and Lin Yan. Exploring data scaling trends and effects in reinforcement learning from human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2503.22230.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2310.03716.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.05316.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/abs/2009.10795.

Zelin Tan, Hejia Geng, Xiaohang Yu, et al. Scaling behaviors of LLM reinforcement learning post-training: An empirical study in mathematical reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. URL https://arxiv.org/abs/2509.25300.

Yves Tillé. *Sampling Algorithms*. Springer Series in Statistics. Springer, New York, 2006.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. https://github.com/huggingface/trl, 2020.

Jiamian Wang, Samyadeep Basu, Koustava Goswami, Tong Yu, and Zhiqiang Tao. Spurious advantage hidden in GRPO. *arXiv preprint arXiv:2609.04063*, 2026a.

Li Wang, Xiaodong Lu, Xiaohan Wang, Yikun Ban, Jiajun Chai, Wei Lin, Tianhao Peng, and Guojun Yin. When self-belief misleads: Active label acquisition for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2605.25864*, 2026b.

Xuekang Wang, Zhuoyuan Hao, Shuo Hou, Hao Peng, Juanzi Li, and Xiaozhi Wang. Reproducing, analyzing, and detecting reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2606.04923*, 2026c.

Yubo Wang, Xueguang Ma, Ge Zhang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.01574.

Jiahao Wu, Ning Lu, Shengcai Liu, Kun Wang, Yanting Yang, Baijiong Lin, Chen Jason Zhang, Li Qing, and Ke Tang. Train at moving edge: Online-verified prompt selection for efficient RL training of large reasoning model. *arXiv preprint arXiv:2603.25184*, 2026a.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 297–312, 2025. URL https://arxiv.org/abs/2307.03025.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, et al. Reasoning or memorization? Unreliable results of reinforcement learning due to data contamination. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026b. URL https://arxiv.org/abs/2507.10532.

Esther Xin. Are verifier errors independent within a GRPO group? Evidence from Qwen2.5 rollouts. *arXiv preprint arXiv:2609.06386*, 2026.

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. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2510.04996.

Zhangchen Xu, Yuetai Li, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. TinyV: Reducing false negatives in verification improves RL for LLM reasoning. *Transactions on Machine Learning Research (TMLR)*, 2026. URL https://arxiv.org/abs/2505.14625.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Minglai Yang, Xinyu Guo, Utkarsh Tyagi, Mian Zhang, Razvan Dumitru, Sunjie Hou, Yunzhong He, Daniel Yue Zhang, and Ying Liu. Rubric dropout: A simple way to mitigate reward hacking in rubric-as-reward RL. *arXiv preprint arXiv:2608.11669*, 2026a.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. GradAlign: Gradient-aligned data selection for LLM reinforcement learning. In *Conference on Language Modeling (COLM)*, 2026b. URL https://arxiv.org/abs/2602.21492.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Sharon Li, Haobo Wang, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, and Gang Chen. Can LLMs learn to reason robustly under noisy supervision? *arXiv preprint arXiv:2604.03993*, 2026c.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.02391.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. Justice or prejudice? Quantifying biases in LLM-as-a-Judge. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.02736.

Juntao Yu. The filter metric is safety-critical: Phantom advantages in group-relative RL under shaped rewards. *arXiv preprint arXiv:2609.13866*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2503.14476.

Yuanqiang Yu, Yanzhao Zheng, Zhentao Zhang, Tianze Xu, Chao Ma, Jihuai Zhu, Jiashun Liu, Xinle Deng, Baohua Dong, Hangcheng Zhu, and Ruohui Huang. PAC: Progress-augmented advantage curriculum for multi-task reinforcement learning of LLMs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. URL https://arxiv.org/abs/2608.30528.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Ilia Kulikov, Kyunghyun Cho, Dong Wang, Yuandong Tian, Jason E Weston, and Xian Li. NaturalReasoning: Reasoning in the wild with 2.8M challenging questions. *arXiv preprint arXiv:2502.13124*, 2025.

Chuyifei Zhang. When the reward suite is leaky: A preregistered causal contrast of natural verifier false positives in RLVR. *arXiv preprint arXiv:2607.11022*, 2026.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning (ICML)*, pp. 1–9, 2015. URL https://arxiv.org/abs/1401.2753.

Yulai Zhao, Haolin Liu, Dian Yu, Sunyuan Kung, Meijia Chen, Haitao Mi, and Dong Yu. One token to fool LLM-as-a-Judge. *arXiv preprint arXiv:2507.08794*, 2025.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. URL https://arxiv.org/abs/2506.02177.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.05685.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://arxiv.org/abs/2410.07137.

Chenyu Zhou. More convincing, not more correct: Self-play reward hacking of reference-free LLM judges. *arXiv preprint arXiv:2607.05904*, 2026.

Yang Zhou, Can Jin, Zihan Dong, Zhepeng Wang, Yanting Yang, Shiyu Zhao, Lei Li, Runxue Bao, Yaochen Xie, and Dimitris N. Metaxas. DARE: Difficulty-adaptive reinforcement learning with co-evolved difficulty estimation. *arXiv preprint arXiv:2605.09188*, 2026.

Haodong Zhu, Yangyang Ren, Yanjing Li, Mingbao Lin, Linlin Yang, Xuhui Liu, Xiantong Zhen, Haiguang Liu, and Baochang Zhang. Unbiased dynamic pruning for efficient group-based policy optimization. In *International Conference on Machine Learning (ICML)*, 2026a. URL `https://arxiv.org/abs/2603.04135`.

Haodong Zhu, Yangyang Ren, Yanjing Li, Sheng Xu, Haiguang Liu, Linlin Yang, and Baochang Zhang. Kalman meets curriculum: Efficient dynamic prompt selection for adaptive RL finetuning. *arXiv preprint arXiv:2607.27610*, 2026b.

Yuxuan Zhu and Daniel Kang. Noisy data is destructive to reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2603.16140*, 2026.

A Extended related work

This appendix extends Section 7. It describes each line of work in more detail and states how our setting differs from it. Our two changes relate to different literatures: the visit bound with verification of the prompts that exceed it relates to bounded reuse and selective labeling (Appendix A.5), and importance-weighted selection to importance weighting and unbiased pruning (Appendix A.6), which ends with a comparison table (Table 3).

A.1 Prompt selection for RLVR

History-based selectors. GRESO (Zheng et al., 2025a) skips a prompt with a probability that grows with the number of consecutive zero-variance groups it has produced and reports up to 2.4 times faster rollout and up to 2.0 times faster training than dynamic sampling. MoPPS (Qu et al., 2026a) keeps a Beta posterior over each prompt's pass rate, draws from it and keeps the prompts whose draws lie nearest a target rate; it reports 78% fewer rollouts than dynamic sampling on Countdown at comparable accuracy. DPPO (Zhu et al., 2026a) scores each prompt by the mean absolute advantage of its previous epoch, prunes a fixed fraction of the lowest-scoring half of each batch, and rescales the retained prompts by the inverse of their keep probability; with completion pruning and dense prompt packing it reports a 2.37-fold reduction in GPU-hours over GRPO for Qwen3-4B on MATH, against 2.31-fold for GRESO in the same table. Two recent selectors track each prompt's logit success rate with a Kalman filter, reporting 71%, 87% and 83% fewer rollouts than dynamic sampling on three policies (Zhu et al., 2026b), or reuse prompts that were informative before together with their neighbors in representation space (Liang et al., 2026b); GPS fits a small generative model of prompt difficulty to the shared optimization history (Qu et al., 2026b). Prompt Replay (Baroian & Berger, 2026) keeps a buffer of prompts whose pass rate in the previous step lay between 0.25 and 0.75. It replays them with fresh completions after a cooldown of 10 steps, with at most 15 reuses. A related method, DARE (Zhou et al., 2026), co-evolves difficulty estimates by self-normalized importance sampling and samples prompts from a Beta distribution over difficulty; its importance weights estimate difficulty and do not reweight the gradient, so it is a selector rather than an inverse-probability correction. All of these methods decide which prompts to roll out from rewards collected in earlier steps, which is the property that lets a response-dependent judge error steer them.

Other selectors and curricula. Online difficulty filtering (Bae et al., 2026), learnability-based selection (Foster et al., 2025) and a curriculum on group-reward variance (Jiang et al., 2025) pick prompts of intermediate difficulty from the rollouts of the current step and keep no per-prompt history, and other methods reallocate rollouts within a step toward hard prompts (Xiong et al., 2026; Noukhovitch et al., 2026). Further methods fit a model of each prompt's difficulty, such as a bandit over problem categories scored by absolute advantage (Chen et al., 2026), a value model trained on past rewards (Gao et al., 2026), attention-based similarity to a small reference set that is rolled out (Sun et al., 2025), a coarse filter on historical reward trajectories followed by an entropy criterion (Wu et al., 2026a), Beta priors from an anchor model updated by training outcomes (Sha et al.,

2026), and a Thompson-sampling allocation across tasks (Yu et al., 2026). ScaleRL removes every prompt whose pass rate reaches 0.9 from all later epochs (Khatri et al., 2026), and in deep reinforcement learning, prioritized level replay revisits environments in order of their temporal-difference errors (Jiang et al., 2021); a recent survey organizes these methods by mechanism (Gholipour et al., 2026). All of these methods, like those above, were evaluated with rule-based or environment rewards. GradAlign (Yang et al., 2026b) selects problems whose policy gradients align with those of a small trusted validation set and is evaluated under unreliable rewards; it needs such a set, which our correction does not. Current-step filters do not accumulate a prompt's past verdicts, which is consistent with our finding that dynamic sampling matches uniform sampling in Science while GRESO and MoPPS do not (Section 3).

Selection with model-based rewards. Two methods tie the choice of prompts to a model-based reward. LLM-as-a-Tutor (Kim et al., 2026) lets the judge compare rollouts pairwise, flag prompts that are no longer challenging and append constraints to them; it acts on the current rollouts and does not study the judge's errors. Pre-PPO (Shen et al., 2025) keeps, once and offline, the prompts with the lowest reward-model scores from an initial PPO run, after observing that high-scoring prompts showed more reward hacking; it is the partial match in Table 3, since it uses reward scores from one earlier run and links the choice of prompts to reward hacking without studying how selection interacts with the reward's errors. Both select prompts once or from the current step; neither follows a prompt's verdicts across visits, which is where we find that a judge's response-dependent errors accumulate.

Comparisons of data policies. DataFlex-RL (Liang et al., 2026a) compares thirteen configurations (two controls, four selection, four reweighting and three mixture policies) over twelve matched seeds with Qwen2.5-7B-Base and rule-based rewards, and finds that none of the eight selection or reweighting methods shows a reproducible gain over uniform sampling. Its selection policies choose responses after rollout and include a stricter DAPO-style filter, but none chooses prompts before rollout from their reward history, so GRESO, MoPPS and DPPO are outside its scope. Its null result concerns accuracy under exact rewards, whereas our harm appears in strict mastery, reward-hacking events and transfer accuracy under a judge.

A.2 Learning progress and unlearnable noise

Burda et al. (2019a) showed that agents rewarded for the error of a learned forward model are drawn to stochastic observations they cannot learn, the noisy-TV problem, and random network distillation (Burda et al., 2019b) removes one source of this attraction by regressing on a deterministic target. Learning-progress criteria reward improvement rather than error and therefore ignore noise that cannot be learned (Oudeyer et al., 2007; Hou et al., 2026). A response-dependent judge error is the opposite case. The policy can learn to earn the judge's credit on a wrong answer, so the judged pass rate on such a prompt rises with training, and the judged progress is real even though the true pass rate stays flat. A selector that favors progress, such as PAC (Yu et al., 2026) and other progress-based curricula, would therefore prefer these prompts; GRESO and MoPPS select on the variance of the next group instead. The growing gap in Figure 2(b) is consistent with the linear accumulation of Proposition 3.

A.3 Imperfect verifiers and judges

Noise models and corrections. Cai et al. (2025) assume noise rates that depend only on the latent correctness and a flip that, given correctness, is independent of the policy's sampling, and they report a stress test with format-dependent, deterministic noise in which their corrections still help. Noise-corrected GRPO (El Mansouri et al., 2026) allows the flip rates to depend on the prompt but draws the flip independently of the response; its experiments estimate one global pair of rates on a held-out balanced subset and gain up to 6.7 points in mathematics with real reward models, and its GRPO variant normalizes advantages by a noise-corrected variance estimate. Rad et al. (2026) model false positives and false negatives in a bandit over recurring reasoning modes and show a phase transition in Youden's index $1 - \alpha - \beta$. In our notation (Section 2), noise-corrected GRPO and Rad et al. (2026) treat the response-level acceptance $a_i(y)$ as a constant α_i. Cai et al. (2025) identify the covariance term that response dependence adds, which is the term C_i of Proposition 3,

and test response-dependent noise under uniform sampling; none of the three studies selection. In the language of label noise, the judge's errors are instance-dependent, and instance-dependent noise is not identifiable from a single noisy label per instance (Liu et al., 2023). Spurious and even random rewards can improve some Qwen models (Shao et al., 2025), which is one reason we measure harm against an exact verifier rather than against training reward.

Systematic and response-dependent errors. Egashira et al. (2026) inject systematic false positives and false negatives into arithmetic RLVR and find delays (false negatives) and plateaus or collapse (false positives), determined by the pattern of the errors rather than their overall rate. Zhang (2026) studies natural false positives of MBPP's own test suites under plain GRPO with 1B to 1.5B policies. The verifier accepts the same wrong programs every time they appear, and the false-positive share stays at its long-run level from the first step and does not grow over 400 to 800 steps (a non-detection its author describes as underpowered). This result serves as our baseline, since it shows that response-dependent errors are not by themselves amplified under plain GRPO; history-based selection is what makes them grow. Xin (2026) measures a within-group correlation of verifier errors of 0.530 on fixed Qwen2.5-1.5B rollouts, without training. Plesner et al. (2026) find that noise rates up to 15% keep peak validation accuracy within 2 points of the clean baseline under uniform sampling, a result they report as consistent across controlled and model-based noise types; Section 4 shows why response-independent noise of this kind produces few reward-hacking events. Yang et al. (2026c) separate wrong labels that the policy can realize, which training reinforces, from those it cannot, and Zhu & Kang (2026) find that training on truly incorrect annotations performs 8–10% worse than training on clean data on mathematical benchmarks. On the verifier side, Xu et al. (2026) show that rule-based checkers produce false negatives, and Huang et al. (2026) that model-based verifiers can be hacked during training.

Supervised analogues and theory. In supervised learning, points selected for high loss or gradient norm are disproportionately mislabeled; RHO-LOSS (Mindermann et al., 2022) avoids them with a model trained on held-out data, which is the independent signal that Observation 1 shows a single judge cannot supply. The hard-to-learn region of a dataset map is rich in mislabeled examples (Swayamdipta et al., 2020), whereas emphasizing uncertain rather than hard samples is more robust to them (Chang et al., 2017). Observation 1 is related to the non-identifiability of common-mode evaluator error from internal judge scores (Afrin & Shihab, 2026) and to the corrupted-reward-channel result of Everitt et al. (2017). The finite-group identity behind Proposition 3 is due to Davis & Recht (2025) for exact binary rewards, with population statistics in Mroueh (2025).

A.4 Judge and rubric exploitation

Optimizing a learned reward overfits its errors as the policy moves (Gao et al., 2023), and ensembles of reward models mitigate but do not remove this because their errors are shared (Eisenstein et al., 2024). Policies exploit model-based verifiers (Huang et al., 2026), rubric judges (Mahmoud et al., 2026) and reference-free judges (Zhou, 2026), and extensional verifiers admit shortcut solutions (Helff et al., 2026). Generic openers and even single punctuation tokens elicit false positives from LLM judges repeatably (Zhao et al., 2025), which is response-dependent error in its plainest form. Mahmoud et al. (2026) report that exploitation grows over training and concentrates in recurring failures, and that stronger verifiers substantially reduce but do not eliminate it, which agrees in direction with our judge-strength comparison (Table 2c). Zhou (2026) observes the judge's pass rate rise from 0.72 to 0.94 while true accuracy stays at 0.20, and Kenton et al. (2026) log verifiable correctness beside the judge, as we do, and find that debate training reduces reward hacking. Proposed remedies change the rubric (Yang et al., 2026a), the judge or its training game (Kenton et al., 2026), or regularize the policy (Ackermann et al., 2026); we locate an amplifier in the data-selection layer and remove it without changing the judge, either by bounding reuse and verifying the prompts that exceed the bound or by reweighting the selector's draws. Yu (2026) shows that filtering groups on a shaped training score lets all-fail groups pass and turns shaping differences into full-size advantages; this is the one-step, deterministic counterpart of Proposition 1, without the history that makes our errors accumulate. The term "spurious advantage" of Wang et al. (2026a) refers to a different effect, and our spurious-group share is unrelated to it.

Style and verbosity bias, and the Science mechanism. LLM judges are known to reward surface features rather than content. A verbosity attack that repeats list items fools Claude-v1 and GPT-3.5 in 91.3% of cases (Zheng et al., 2023), and changing only a verbosity instruction moves a model's AlpacaEval win rate from 22.9% to 64.3% (Dubois et al., 2024). Weak judges prefer the longer but worse answer on most items of a length-bias benchmark (Park et al., 2024), and judges rate answers with minor factual errors above correct but short ones (Wu & Aji, 2025). Content-free responses such as a single colon or "Thought process:" are accepted as correct by open-weight judges at per-dataset average rates of 66% to 89% for LLaMA3-70B-Instruct (Zhao et al., 2025). Our rubric judge's acceptance of 82% of responses that only restate the question (Appendix C) is comparable to these rates for open-weight judges, whereas frontier judges accept far fewer content-free responses (Zhao et al., 2025) and are more robust to verbosity (Ye et al., 2025).

Policies trained against such rewards learn the rewarded feature. Length explains 70% to 90% of the reward improvement in two RLHF settings (Singhal et al., 2024), and the most common reward hacking in practice is verbosity (Chen et al., 2024). Rubric-trained policies learn to match the topic of criteria rather than their precise claims (Mahmoud et al., 2026), injected judge biases are picked up within about a hundred steps (Wang et al., 2026c), and RL against a small model verifier discovers single-symbol exploits (Huang et al., 2026). The learned feature is whatever the judge rewards: usually length for preference reward models, and for our rubric judge terse final answers that restate the question and the rubric's terms. The mechanism we report for Science adds selection to this picture. A history-based selector keeps the prompts whose groups stay split; splits that follow style persist, while splits that follow correctness resolve as the policy learns, so the selector's batches carry a larger share of style-split groups than uniform sampling's or dynamic sampling's (Section 3). This is amplification by composition rather than by revisiting individual prompts, which is why it appears before any prompt is visited unusually often, and why it shrinks with a stronger judge, which splits groups less on style.

A.5 Bounded reuse and targeted verification

Two history-based selectors already limit how often a prompt is reused. Prompt Replay (Baroian & Berger, 2026) blocks a replayed prompt for a cooldown of C steps and allows at most R reuses before the prompt returns to the ordinary sampler, and its authors describe larger reuse and shorter cooldowns as trading speed against a higher risk of overfitting. LEEPS (Liang et al., 2026b) warns that repeatedly exploiting historically informative prompts can narrow training coverage. Both limits are set against overfitting under exact rewards, and neither is tied to verification. On the labeling side, active label acquisition for RLVR (Wang et al., 2026b) buys ground-truth labels for a few samples chosen by the expected change they make to the advantage and trains on pseudo-labels elsewhere, and importance-weighted active learning (Beygelzimer et al., 2009) decides which labels to query by the disagreement of candidate hypotheses.

Our visit bound differs in its purpose and in what it triggers. It limits the share of the rollout budget any prompt may absorb because, under a judge, the prompts that a history-based selector revisits most are where the judge's response-dependent errors accumulate (Section 3), and crossing the bound is the event that sends a prompt to independent verification. In Math the exact verifier checks only those prompts and retires the ones on which the judge accepted a wrong answer, which costs 416 audits over 9,600 trained groups, or 4.3% additional verifier calls. It brings GRESO's judged-minus-true gap from 0.129 to 0.012, below uniform sampling's 0.040 (Section 5). The bound alone requires no judge information. Neither Prompt Replay's reuse cap nor label acquisition by expected value selects the prompts to verify by how often selection has returned to them, which is the quantity that tracks the judge's errors.

A.6 Importance weighting and unbiased pruning

The weight $P/(N\rho_{t,i})$ is the Horvitz–Thompson estimator (Horvitz & Thompson, 1952), and we summarize its cost with Kish's effective sample size (Kish, 1965). Importance-weighted active learning (Beygelzimer et al., 2009) queries labels with floored probabilities and weights each queried example by the inverse of its query probability, which is the mechanism of our wrapper applied to label queries rather than prompt draws. Importance-sampling SGD draws examples non-uniformly and reweights them so that the gradient stays unbiased (Needell et al., 2014; Zhao & Zhang, 2015;

Alain et al., 2016; Katharopoulos & Fleuret, 2018; Csiba & Richtárik, 2018), and the variance-minimizing design behind Theorem 1 goes back to Godambe & Joshi (1965). In RLVR, VIP (Nguyen et al., 2026) and GVM-RAFT (Yao et al., 2025) allocate rollouts across prompts to minimize gradient variance under a budget, with exact rewards; and prioritized experience replay corrects its sampling bias with weights $(NP(i))^{-\beta}$ that compensate fully when $\beta = 1$ (Schaul et al., 2016). InfoBatch (Qin et al., 2024) prunes low-loss examples at random, using the losses of the previous epoch, and rescales the retained low-loss ones by $1/(1-r)$ to keep the expected update.

Relation to DPPO. DPPO (Zhu et al., 2026a) applies the same two-level rule to prompts and completions in GRPO. Prompts in the lowest-scoring half are pruned with rate r_q (deterministically, as a fixed fraction of each candidate set), prompt pruning is skipped in the first epoch, the retained prompts are rescaled by the inverse keep probability, and a separate theorem bounds the variance this adds. Our correction coincides with DPPO's prompt-level rescaling up to a constant factor, since DPPO's weight also carries the kept fraction of the batch. It differs in that it wraps the keep propensities of an existing selector instead of introducing a new score, draws a fixed number of prompts from the whole pool at every step so that a prompt that stays informative is revisited more often at a proportionally smaller weight, is exact for the token-normalized loss once the normalizer is fixed before the draw, and is analyzed under judge error. Section 5 adds the bound on the savings that any unbiased design can keep (Theorem 1). Under our Math judge, DPPO, which thins uniformly drawn batches, stays at uniform sampling's level of reward-hacking events, as does IW-GRESO, while GRESO does not (Table 2b), as expected for any design that keeps each prompt's expected weighted exposure at uniform sampling's level (Proposition 4). DPPO reaches this safety with 0.59 to 0.60 of uniform sampling's rollouts and a higher per-step second moment, which is consistent with Theorem 1, whereas our wrapper draws P prompts at every step and keeps no rollout savings of its own; its use is to make an existing selector safe without changing the selector's scoring rule. Horvitz–Thompson weights also appear at the level of tokens (Sang et al., 2026) and in replays with recorded inclusion probabilities for rubric-based credit assignment (Hu, 2026); neither selects prompts.

Table 3: Related methods and studies. The columns mark whether the method selects prompts or data from their reward or loss history, whether it was evaluated with a model-based reward (LLM judge or reward model), whether it models or measures reward errors that depend on the response, whether it corrects selection bias with inverse-probability weights, and whether it studies prompt selection under reward error. (✓) marks a partial match, explained in the text.

	reward or loss history	model-based reward	response-dependent errors	inverse-probability weights	selection under error
Dynamic sampling (Yu et al., 2025)	–	–	–	–	–
GRESO, MoPPS, KGPS, LEEPS, Prompt Replay	✓	–	–	–	–
DARE (Zhou et al., 2026)	✓	–	–	–	–
DPPO (Zhu et al., 2026a)	✓	–	–	✓	–
InfoBatch (Qin et al., 2024)	✓	–	–	✓	–
Importance-weighted active learning (Beygelzimer et al., 2009)	–	–	–	✓	–
Pre-PPO (Shen et al., 2025)	(✓)	✓	–	–	(✓)
LLM-as-a-Tutor (Kim et al., 2026)	–	✓	–	–	–
Noise-corrected GRPO (El Mansouri et al., 2026)	–	✓	–	–	–
Cai et al. (2025)	–	–	✓	–	–
Plesner et al. (2026)	–	✓	–	–	–
Egashira et al. (2026)	–	–	✓	–	–
Zhang (2026)	–	–	✓	–	–
Mahmoud et al. (2026), Zhou (2026)	–	✓	✓	–	–
This work	✓	✓	✓	✓	✓

B NOTATION, SELECTORS AS SAMPLING DESIGNS, AND METRICS

This appendix collects the notation (§B.1), states the loss exactly as it was trained (§B.2), writes every selection rule we run as a sampling design (§B.3), reports cumulative and per-step amplification (§B.4), and defines every metric with the rule we apply when a run ends before a quantity is observed (§B.5).

B.1 NOTATION

Table 4 lists the symbols used throughout the paper. Indices are i for prompts, j for the responses of a group, ℓ for tokens and t for training steps. Quantities that depend on the policy carry the argument θ where the dependence matters.

Table 4: Notation.

Symbol	Meaning
Training	
$\mathcal{D} = \{x_i\}_{i=1}^{N}$	prompt pool; $N = 800$ (Math) and $N = 406$ (Science)
T, t	number of training steps (300 in Math, 200 in Science) and step index
B_t, P	batch of prompts trained at step t and its size, $P = 32$
P_r	prompts rolled out per step ($P_r = P$, except $P_r = 48$ for dynamic sampling)
G; y_{ij}; r_{ij}	group size ($G = 8$); response j to prompt i; its reward
$\hat{A}_{ij}$	advantage $(r_{ij} - \bar{r}_i)/(s_i + 10^{-4})$, with $\bar{r}_i$, s_i the group mean and sample standard deviation
$\eta_{ij\ell}(\theta)$	likelihood ratio of token ℓ of y_{ij} between the current and the sampling policy
$\epsilon_{\text{low}}, \epsilon_{\text{high}}$	clip range, 0.2 and 0.28
ν_{ij}	sequence-level ratio of trainer to sampler likelihood (sampler-mismatch correction, masked above 3)
Z_t	loss normalizer (completion tokens; fixed before the draw in the importance-weighted arms)
Selection	
$\mathcal{F}_t$	history of draws and verdicts before step t
$\kappa_{t,i}$	keep propensity that a selector assigns to prompt i
$\rho_{t,i}$	inclusion probability $\Pr(i \in B_t \mid \mathcal{F}_t)$
$w_{t,i}$	weight on group i's advantages (1, or $P/(N\rho_{t,i})$ in the importance-weighted arms)
$n_{t,i}$	number of visits of prompt i up to and including step t
Λ	amplification (weighted exposure), $\max_i \sum_{t \le T} \rho_{t,i} w_{t,i}/(TP/N)$
Λ_{vis}	visit amplification, $\max_i n_{T,i}/(TP_r/N)$; $\Lambda_{\text{vis}}(t)$ uses t in place of T
Λ_t	per-step amplification $\max_i \rho_{t,i}/(P/N)$
$c_{t,i}$; $\phi_{\text{easy}}, \phi_{\text{hard}}$	GRESO's zero-variance streak (consecutive zero-variance groups); its base keep rates
$\Lambda_{\max}$	visit-cap multiplier (a prompt is skipped while $n_{t,i} > \Lambda_{\max} tP/N$)
Judge	
J, V	judge verdict and exact-verifier verdict of a response, in $\{0, 1\}$
k^{obs}, k^{true}	number of responses in a group accepted by the judge and by the exact verifier
$p_i(\theta)$	true pass rate of prompt i
$a_i(y)$, $b_i(y)$	probability that the judge accepts the wrong response y, or rejects the correct response y
$\alpha_i(\theta)$, $\beta_i(\theta)$	false-positive and false-negative rates, $\mathbb{E}_{y\sim\pi_\theta}[a_i(y) \mid V{=}0]$ and $\mathbb{E}_{y\sim\pi_\theta}[b_i(y) \mid V{=}1]$
$q_i(\theta)$	judged pass rate $p_i(1-\beta_i) + (1-p_i)\alpha_i$
Sets and metrics	
$\mathcal{H}$	hard set, the 359 Math prompts that the base policy solves at most twice in eight attempts
$\hat{\mathcal{E}}$	error-prone set, the 106 Math prompts on which $k^{\text{obs}} > k^{\text{true}}$ in at least half of their uniform-sampling visits
δ	judged-minus-true gap, the mean of $(k^{\text{obs}} - k^{\text{true}})/G$ over trained groups

Section 4 and Section 5 add the GRPO weight $\omega_G(q)$, the judge error term $e_i = q_i - p_i$, the bias bound Δ on an unknown set $\mathcal{E}$ of at most m prompts, which, unlike $\hat{\mathcal{E}}$, is not measured, and the per-prompt second moment $M_{t,i} = \mathbb{E}[\|g_i\|^2 \mid \mathcal{F}_t]$. The symbol s_i always denotes the group's sample standard deviation; the score function is written $\nabla_\theta \log \pi_\theta$.

B.2 The loss as trained

All runs use TRL 1.9.2 (von Werra et al., 2020) with the loss type `dapo` and no KL term, so there is no reference model. The advantage divides by the group's sample standard deviation (with Bessel's correction) plus 10^{-4}. For any informative group $\sum_j \hat{A}_{ij}^2 = (G-1)\, s_i^2/(s_i + 10^{-4})^2$, which is $G - 1 = 7$ up to the stabilizer. A Math group with two correct responses out of eight has advantages $+1.620$ and -0.540 and a sum of squares of 6.997. The loss of step t is

$$\mathcal{L}_t(\theta) = -\frac{1}{Z_t} \sum_{i \in B_t} w_{t,i} \sum_{j=1}^{G} \sum_{\ell=1}^{|y_{ij}|} \min\big(\eta_{ij\ell} \hat{A}_{ij},\ \mathrm{clip}(\eta_{ij\ell}, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})\, \hat{A}_{ij}\big), \tag{2}$$

with $(\epsilon_{\mathrm{low}}, \epsilon_{\mathrm{high}}) = (0.2, 0.28)$, so the clipped token-level surrogate is summed over every token of every response in the batch and divided by Z_t.

The clip never binds. Each batch of P groups is generated once and used for exactly one optimizer step, since the gradient-accumulation micro-batches together make up the generation batch, so the policy that computes the loss is the policy that generated the batch. The ratio $\eta_{ij\ell}$ therefore equals 1 wherever the gradient is taken, the gradient of the clipped surrogate equals that of the unclipped one, and the logged fraction of clipped tokens is zero at every step. The clip range $(0.2, 0.28)$ of DAPO (Yu et al., 2025) is part of the configuration but has no effect on any update in this paper.

Sampler-mismatch correction. Responses are generated by a vLLM engine colocated with the trainer, and trainer and sampler are both run in fp16. TRL corrects the remaining numerical mismatch with masked sequence-level importance weighting, in which each response's token losses are multiplied by $\nu_{ij} = \exp \sum_\ell \big[\log \pi_\theta^{\mathrm{train}}(y_{ij\ell} \mid \cdot) - \log \pi_\theta^{\mathrm{sample}}(y_{ij\ell} \mid \cdot)\big]$ and set to zero when $\nu_{ij} > 3$. The ratio stays close to one, with a mean per step between 0.9995 and 1.0007 and the largest logged ratio is 2.99. No response was masked in any Math run or in any Science run of the recipe used in Sections 3 and 6; masking occurred on 14 logged steps in all, every one of them in Science runs of the training recipe without the overlong penalty (Appendix F). Because ν_{ij} acts on single responses and $w_{t,i}$ on whole groups, the two weights commute, and the correction does not interact with selection.

The normalizer. For uniform sampling, GRESO and MoPPS, Z_t is the realized number of completion tokens in the P groups of step t, zero-variance groups included. Dynamic sampling divides by the completion tokens of all $1.5P$ groups it rolls out, including the ones it discards, which keeps the implicit rescaling of the filter constant across steps instead of letting it drift with the fraction of informative groups; its steps are therefore about two thirds the size of a normalization by the trained groups alone, a constant factor that Adam largely absorbs. The importance-weighted arms fix Z_t before the draw, as an exponential moving average (rate 0.2) of earlier steps' Horvitz–Thompson-weighted token counts (Horvitz & Thompson, 1952); only at step 0, which has no history, is the step's own weighted count used. Proposition 4 needs this $\mathcal{F}_t$-measurable normalizer. Noise-corrected GRPO, applied to GRESO in Section 6, normalizes advantages by its own noise-corrected standard deviation instead of s_i (Appendix E).

Rewards. In Math the reward is the judge's binary verdict on the full response, and a verdict that cannot be parsed counts as 0; the exact verifier's verdict is logged next to it and never enters the loss. Completions are capped at 1,024 tokens. In Science a response passes the rubric if every Essential criterion is met, every Pitfall is avoided, and the weighted score $\sum_c |u_c|\, \mathrm{met}_c / \sum_c |u_c|$ (with rubric weights u_c) over the rubric's criteria is at least $\tau = 0.5$. The reward is the pass indicator plus DAPO's soft overlong penalty. The penalty is 0 up to 1,536 tokens and falls linearly to -1 at the 2,048-token cap. The penalty gives an all-fail group with unequal lengths a nonzero advantage, so selectors and dynamic sampling read informativeness from the pass indicators rather than from

the shaped reward, and a zero-variance prompt is one whose G pass indicators are equal. Optimizer settings are listed in Table 19.

B.3 SELECTORS AS SAMPLING DESIGNS

A selection rule is a sampling design. At step t it chooses B_t at random given $\mathcal{F}_t$, which fixes the inclusion probabilities $\rho_{t,i}$, and it attaches a weight $w_{t,i}$ to each chosen group. By linearity, a design's expected update given $\mathcal{F}_t$ is uniform sampling's update with prompt i reweighted by $N\rho_{t,i}w_{t,i}/P$ (Section 4), so the exposure $X_{t,i} = \rho_{t,i}w_{t,i}$ is the quantity that matters, and the amplification Λ compares its largest cumulative value with uniform sampling's TP/N. We take $\mathcal{F}_t$ to contain the past draws and verdicts and the current epoch's permutation of the pool. Table 5 summarizes the designs we run; the paragraphs below give the details.

Table 5: The selection rules of this paper as sampling designs. The last two columns give the visit amplification Λ_{vis} as measured at the end of training (Math, then Science; mean $\pm$ SE over three seeds where it varies across seeds). For the weighted arms the weighted exposure is $\Lambda = 1$ exactly, while their draws, which are independent across steps, give the visit amplification shown.

Rule	Keep propensity $\kappa_{t,i}$ and draw	Weight $w_{t,i}$	Λ_{vis}, Math	Λ_{vis}, Science
Uniform	$\kappa \equiv 1$; epoch-wise permutation, P prompts per step	1	1.00	1.08
Dynamic sampling	$\kappa \equiv 1$; $1.5P$ prompts per step from the permutation	1 for up to P judge-informative groups, 0 otherwise	1.06	1.06
GRESO	$\max(\phi^{c_{t,i}}, 0.01)$ after a streak of $c_{t,i}$ zero-variance groups; scan until P are kept	1	11.40 ± 0.10	5.96 ± 0.22
MoPPS	top P of $\min(16P, N)$ candidates by $\lvert\tilde{q}_{t,i} - 0.5\rvert$, with $\tilde{q}_{t,i}$ drawn from a Beta posterior	1	13.18 ± 0.09	9.16 ± 0.22
Visit cap	inner rule, skipping i once $n_{t,i} \geq 8$ and $n_{t,i} > \Lambda_{\max} \max(1, tP/N)$; freed places filled uniformly	1	1.50 at $\Lambda_{\max} = 1.5$	–
Floored GRESO, unweighted	GRESO's κ floored at 0.3; $\rho_{t,i} \propto \kappa_{t,i}$ with $\sum_i \rho_{t,i} = P$; systematic PPS draw	1	3.05 ± 0.07	2.24 ± 0.03
IW-GRESO	the floored GRESO draw	$P/(N\rho_{t,i})$, so $\Lambda = 1$	3.00 ± 0.10	2.20 ± 0.04
IW-MoPPS	MoPPS's band mass floored at 0.3; same draw	$P/(N\rho_{t,i})$, so $\Lambda = 1$	3.42 ± 0.05	2.71 ± 0.08

Uniform sampling. In the theory, uniform sampling is the reference design that includes each prompt with probability $\rho_{t,i} = P/N$ at every step, so $X_{t,i} = P/N$ and $\Lambda = 1$. We run it as an epoch-wise permutation of the pool that drops the last partial batch of each epoch. Its marginal inclusion probability is also P/N; given $\mathcal{F}_t$, which contains the permutation, $\rho_{t,i} \in \{0, 1\}$, so its weighted exposure equals its realized visits and its Λ equals its Λ_{vis}. The permutation equalizes realized visits. In Math the pool is exactly 25 batches of 32 prompts. Every prompt is therefore visited 12 times in 300 steps, and $\Lambda_{\text{vis}} = 1.00$ in every seed. In Science the pool of 406 prompts is 12 batches plus 22 prompts, which sit out each epoch. The expected number of visits is $TP/N = 15.76$. The realized counts range from 12 to 17 in every seed, and the largest gives $\Lambda_{\text{vis}} = 1.08$. An independent uniform draw at each step would show $\Lambda_{\text{vis}} \approx 2.00$ in Math and 1.78 in Science by chance alone, which is the reference value for the importance-weighted arms, whose draws are independent across steps.

Dynamic sampling. DAPO resamples until P informative groups are collected (Yu et al., 2025). We run the fixed-budget variant used in our comparisons, in which each step rolls out the next $P_r = 1.5P = 48$ prompts of the epoch permutation once, trains the judge-informative groups first, and fills any remaining places among the P with padding groups whose advantages are set to zero. Each step therefore generates 384 rollouts instead of 256. Λ_{vis} counts every rolled-out prompt and equals 1.06 in both domains. Each epoch drops its last partial chunk of prompts, 32 in Math and 22

in Science. Its weights depend on the verdicts of the current step and are therefore not fixed by $\mathcal{F}_t$; given $\mathcal{F}_t$, every rolled-out prompt has weight at most 1, so its exposure per step is at most $1.5P/N$ for every prompt alike, and its visits stay as even as uniform sampling's.

GRESO. GRESO (Zheng et al., 2025a) keeps two counters per prompt, the number $c^+_{t,i}$ of consecutive most recent groups in which all responses passed and the number $c^-_{t,i}$ of consecutive most recent groups in which none passed (at most one of the two is nonzero). Its keep propensity is

$$\kappa_{t,i} = \begin{cases} \max\big(\phi_{\text{easy}}^{\,c^+_{t,i}}, 0.01\big) & \text{if } c^+_{t,i} > 0, \\ \max\big(\phi_{\text{hard}}^{\,c^-_{t,i}}, 0.01\big) & \text{if } c^-_{t,i} > 0, \\ 1 & \text{otherwise,} \end{cases}$$

and we write $c_{t,i}$ for whichever counter is active and call it the *zero-variance streak*. The base rates start at 0.5 and, from the end of the first epoch, move by 0.01 per step within $[0.05, 0.95]$. They move toward a target share of zero-variance groups of 0.25. The target is split one to two between all-pass groups (0.083) and all-fail groups (0.167). A step scans prompts in the order of the epoch permutation, keeps each with probability $\kappa_{t,i}$, and continues until P prompts are kept; if every prompt of the pool has been examined first, the remaining places are filled with prompts skipped at that step. The inclusion probability is thus $\rho_{t,i} = \Pr(i \text{ examined at } t \mid \mathcal{F}_t)\,\kappa_{t,i}$ up to these forced fills. In our runs ϕ_{easy} reaches its floor of 0.05 at step 68 in Math, the earliest step the tuner allows. Steps are counted from 0, and the tuner's 45 decrements of 0.01 begin after the first epoch. In Science the floor is reached between steps 56 and 62. From then on, a prompt that passed on its last one, two or three visits is kept with probability 0.05, 0.01 and 0.01. The rate ϕ_{hard} rises to 0.94–0.95. The rule is ported from the authors' code, and the rewards it reads are the judge's verdicts.

MoPPS. MoPPS (Qu et al., 2026a) keeps a Beta posterior of each prompt's pass rate, starting from $\text{Beta}(1,1)$ and adding k^{obs} successes and $G - k^{\text{obs}}$ failures after each visit without discounting (the authors' released configuration). At each step it takes $\min(16P, N)$ candidates from the permutation stream (512 in Math, the whole pool in Science), draws $\tilde{q}_{t,i}$ from each candidate's posterior and trains the P candidates with the smallest $(\tilde{q}_{t,i} - 0.5)^2$. Its inclusion probability is the probability that i is a candidate times the probability that its draw ranks among the P nearest to 0.5, which has no closed form. The importance-weighted version therefore uses as $\kappa_{t,i}$ the posterior mass of the pass rate inside the band $[0.3, 0.7]$.

Visit cap and cooldown. A visit cap with multiplier $\Lambda_{\max} \in \{1.5, 2.1\}$ skips prompt i once it has at least 8 visits and more than $\Lambda_{\max} \max(1, tP/N)$ of them, and fills the freed places uniformly; a cooldown makes a prompt ineligible for 12 steps after each visit. Both wrap an inner selector. Because of the 8-visit minimum, a prompt can reach 8 visits while the uniform share tP/N is still 4. This gives the Λ_{vis} of 1.92 at step 100 in Table 6. From step 150 on, when $1.5\,tP/N \geq 9$, the cap binds exactly.

Importance-weighted selection. IW-GRESO and IW-MoPPS read the selector's keep propensity, floor it at 0.3, set $\rho_{t,i} = \min(1, \lambda_t \kappa_{t,i})$ with λ_t chosen by water-filling so that $\sum_i \rho_{t,i} = P$, and draw exactly P prompts by systematic probability-proportional-to-size sampling over a fresh random order of the whole pool, which realizes these inclusion probabilities exactly. Each drawn group's advantages are multiplied by $w_{t,i} = P/(N\rho_{t,i})$ after group normalization, so $X_{t,i} = P/N$ for every prompt and $\Lambda = 1$ exactly, conditional on $\mathcal{F}_t$ and whatever the selector's state. The selector's state is updated from the drawn groups as usual. Section 5 and Appendix D give the estimator, its exactness conditions and the pseudocode.

B.4 Cumulative and per-step amplification

Table 6 traces $\Lambda_{\text{vis}}(t) = \max_i n_{t,i}/(tP_r/N)$ through training. In Math, GRESO cannot skip any prompt during the first epoch and its tuner starts from 0.5, so its amplification over the first 50 steps is identical in every seed; its amplification then grows steadily to 11.40 as the keep rate of passed prompts reaches its floor. MoPPS concentrates from the first steps, since its Thompson draws favor

prompts whose posterior pass rate lies near 0.5, and reaches 8.4 by step 50. Both replays copy the schedule of a GRESO run and therefore share the amplification of the run they copy. In Science, where the pool is half as large, each prompt is due 15.76 visits. At step 50 GRESO's amplification is 1.78 in every seed, while MoPPS's is already 6.09.

Table 6: Cumulative visit amplification $\Lambda_{\mathrm{vis}}(t)$ (mean $\pm$ SE over three seeds; values without $\pm$ are identical in every seed). In Math, GRESO and MoPPS seeds 0 and 1 are the mean of two runs; the last column uses each run's final logged step.

Math ($N = 800$; uniform share $tP/N = 1, 2, 4, 6, 8, 12$)						
Arm	$t = 25$	50	100	150	200	300
Uniform	1.00	1.00	1.00	1.00	1.00	1.00
GRESO	1.00	1.50	3.54 ± 0.04	6.11 ± 0.06	8.23 ± 0.02	11.40 ± 0.10
MoPPS	6.17 ± 0.17	8.42 ± 0.22	10.75 ± 0.07	12.06 ± 0.06	12.61 ± 0.06	13.18 ± 0.09
Both replays	1.00	1.50	3.58 ± 0.08	6.11 ± 0.06	8.29 ± 0.04	11.47 ± 0.05
GRESO + cap 1.5	1.00	1.50	1.92 ± 0.08	1.50	1.50	1.50
Science ($N = 406$; uniform share $tP/N = 1.97, 3.94, 7.88, 11.82, 15.76$; dynamic sampling $\times 1.5$)						
Arm	$t = 25$	50	100	150	200	
Uniform	1.52	1.27	1.14	1.10	1.08	
Dynamic sampling	1.35	1.18	1.10	1.07	1.06	
GRESO	1.52	1.78	2.54 ± 0.07	4.20 ± 0.17	5.96 ± 0.22	
MoPPS	5.58 ± 0.29	6.09 ± 0.14	7.78 ± 0.11	8.77 ± 0.15	9.16 ± 0.22	

The per-step amplification $\Lambda_t = \max_i \rho_{t,i}/(P/N)$ describes a single draw. For GRESO, a prompt with no active streak is kept whenever it is examined, so Λ_t is close to the number of prompts examined at step t divided by P, and the mean keep rate among examined prompts is $\bar{\kappa}_t = P/(\text{examined})$. Table 7 shows that late in training GRESO examines most of the pool at every step, so Λ_t approaches its ceiling N/P (25 in Math, 12.7 in Science). Uniform's epoch permutation reaches the same ceiling at the last step of every epoch, when the P remaining prompts are drawn with certainty. The per-step maximum is therefore not informative about accumulated exposure, which is why the paper uses the cumulative quantities Λ and Λ_{vis}.

Table 7: Per-step amplification Λ_t of GRESO, mean keep rate $\bar{\kappa}_t$ among examined prompts, and forced fills per step (prompts filled after the whole pool was examined), each averaged over the steps of a 50-step window (mean $\pm$ SE over three seeds; values without $\pm$ are identical in every seed).

	Math			Science		
Steps	Λ_t	$\bar{\kappa}_t$	forced fills	Λ_t	$\bar{\kappa}_t$	forced fills
$[0, 50)$	1.3	0.765 ± 0.008	0.0	1.3	0.780	0.0
$[50, 100)$	5.3	0.188 ± 0.000	0.0	3.2 ± 0.0	0.312 ± 0.004	0.0
$[100, 150)$	10.6 ± 0.1	0.094 ± 0.001	0.3 ± 0.0	7.4 ± 0.1	0.135 ± 0.002	0.0
$[150, 200)$	14.6 ± 0.1	0.068 ± 0.000	1.0 ± 0.1	12.1 ± 0.1	0.082 ± 0.001	3.5 ± 0.9
$[200, 250)$	19.0 ± 0.1	0.053 ± 0.000	2.8 ± 0.2	–	–	–
$[250, 300)$	22.0 ± 0.2	0.046 ± 0.000	4.8 ± 0.5	–	–	–

B.5 Metric definitions and the censoring rule

Visits and groups. A visit is a step at which a prompt's group is rolled out, and a prompt's visit number is the order of its visits by step, starting at 1. Trained groups are the groups of B_t; for dynamic sampling, visits count every rolled-out prompt, as in Λ_{vis}. A group is *zero-variance* if all its rewards are equal and *informative* otherwise; with the verdict counts k^{obs} and k^{true} it is *judge-informative* if $0 < k^{\mathrm{obs}} < G$ and *truly informative* if $0 < k^{\mathrm{true}} < G$. In Science, where the reward is shaped, these terms refer to the pass indicators.

Hard set. $\mathcal{H}$ contains the prompts of the Math pool that the base policy solves at most twice in eight attempts at 1,024 tokens under the exact verifier, 359 of 800. It is fixed before training and is the same for every arm.

Reward-hacking event. A reward-hacking event is a visit to a prompt of $\mathcal{H}$ with $k^{\text{true}} = 0$ and $k^{\text{obs}} \geq 6$ whose preceding visit was judge-informative with $k^{\text{true}} > 0$; each prompt counts once. Equivalently, visit v is at risk if visit $v - 1$ was judge-informative with $k^{\text{true}} > 0$ and the prompt has had no event before, and the event is the first at-risk visit on which the verifier accepts none of the eight responses while the judge accepts at least six. Events are counted over all logged steps of a run. Under response-independent errors at the Math judge's base false-positive rate $\alpha = 0.22$, a visit whose eight responses are all wrong produces such a group with probability 2.09×10^{-3}, the tail $\Pr(\text{Bin}(G, \alpha) \geq 6)$; Corollary 1 compares the measured counts with the counts expected by chance under response-independent errors with per-prompt rates. Table 8 shows that the comparisons of the paper do not depend on the threshold of six. At every threshold from five to eight, both selectors produce more events than uniform sampling, by at least 6.8 events, several times the standard error of the difference, while the difficulty-matched replay stays at uniform sampling's level.

Table 8: Reward-hacking events on $\mathcal{H}$ (Math, whole run unless stated) as a function of the judge threshold (mean $\pm$ SE over three seeds; paired differences against the same-seed uniform run). For GRESO and MoPPS, whose seeds 0 and 1 have two runs each, the two runs are averaged within the seed first. The paper's definition is the column "≥ 6".

Arm	$k^{\text{obs}} \geq 5$	$k^{\text{obs}} \geq 6$	$k^{\text{obs}} \geq 7$	$k^{\text{obs}} = 8$	$k^{\text{obs}} \geq 6$ by step 200
Uniform	15.3 ± 0.7	11.3 ± 0.3	6.7 ± 0.3	4.7 ± 0.3	7.3 ± 0.3
GRESO	34.3 ± 0.3	29.8 ± 0.2	21.0 ± 0.3	11.5 ± 0.3	20.5 ± 0.5
− uniform	$+19.0 \pm 0.6$	$+18.5 \pm 0.3$	$+14.3 \pm 0.6$	$+6.8 \pm 0.4$	$+13.2 \pm 0.7$
MoPPS	35.3 ± 1.2	32.3 ± 0.7	26.7 ± 0.7	15.8 ± 0.8	24.8 ± 1.1
− uniform	$+20.0 \pm 1.5$	$+21.0 \pm 0.6$	$+20.0 \pm 0.6$	$+11.2 \pm 1.2$	$+17.5 \pm 1.3$
Replayed schedule	33.0 ± 2.1	26.7 ± 1.8	20.3 ± 1.5	12.3 ± 0.9	18.7 ± 1.2
− uniform	$+17.7 \pm 2.0$	$+15.3 \pm 2.0$	$+13.7 \pm 1.8$	$+7.7 \pm 0.9$	$+11.3 \pm 1.5$
Matched replay	15.0 ± 1.7	11.0 ± 1.2	9.0 ± 1.2	3.3 ± 0.3	9.0 ± 1.2
− uniform	-0.3 ± 1.9	-0.3 ± 1.5	$+2.3 \pm 1.5$	-1.3 ± 0.3	$+1.7 \pm 1.5$
GRESO + cap 1.5	24.0 ± 2.9	15.0 ± 1.7	12.0 ± 1.2	6.0 ± 0.6	9.0 ± 1.2
− uniform	$+8.7 \pm 3.0$	$+3.7 \pm 2.0$	$+5.3 \pm 1.5$	$+1.3 \pm 0.7$	$+1.7 \pm 1.5$

Averaging within seed hides the spread between the two runs of a seed, which is larger than the spread between seed means. At the paper's threshold the individual GRESO runs produce 23 to 37 events. The MoPPS runs produce 22 to 44. The uniform-sampling runs produce 11 to 12.

Event rate and chance-adjusted events. Raw event counts grow with exposure, since a selector that revisits a prompt gives it more chances to produce an event. The two primary Math metrics therefore normalize for exposure. The *event rate* is 1,000 times a run's events divided by its at-risk visits to $\mathcal{H}$. The *chance-adjusted event count* is a run's events minus the number expected by chance under response-independent errors. Each of the run's at-risk visits to prompt i is taken to produce an event with probability $\Pr(\text{Bin}(G, \alpha_i) \geq 6)$, with per-prompt false-positive rates α_i drawn from the distribution fitted to the judge-informativeness of all-wrong groups in the uniform runs (Corollary 1, Appendix D), and the probabilities are summed over the run's own at-risk visits. A chance-adjusted count near zero means the events are what the judge's per-prompt error rates produce at the run's exposure; a positive count means events beyond that, which response-dependent errors explain. Raw counts are reported as secondary.

Spurious-group share. The spurious-group share is the fraction of trained groups that are judge-informative but verifier-unanimous ($0 < k^{\text{obs}} < G$ and $k^{\text{true}} \in \{0, G\}$), computed over the last 20% of logged steps. Such groups carry only spurious signal, since their advantages sort the responses by the judge's errors while all responses are equally right or equally wrong. The quantity is distinct from the spurious advantage of Wang et al. (2026a).

Style-split share (Science). Science has no exact verifier, so the reference judge takes its place: DeepSeek-V4-Flash grades the same responses against the reference answer with the training reward

rule, and agrees with the panel on 98.9% of pass verdicts (Appendix C). A trained Science group is *style-split* when the training judge splits it ($0 < k^{\text{obs}} < G$) while the reference judge gives all G responses the same pass verdict. Its split then does not track correctness, and within such groups it follows surface features such as restating the question or rubric terms and a terse final answer without a derivation (Section 3). The style-split share is the fraction of trained groups that are style-split, per window over training or over the last 20% of steps; it is the Science counterpart of the spurious-group share.

Judged-minus-true gap. δ is the mean of $(k^{\text{obs}} - k^{\text{true}})/G$ over trained groups. As a run-level number it is taken over the last 20% of logged steps, that is, the last $\lfloor n/5 \rfloor$ of a run's n logged steps (60 of 300). Figure 2b instead averages it by visit number over each run's prompts with at least 49 visits. The analyses by visit number in Appendix F cut visit numbers at 12, 24 and 48.

Strict mastery. Strict mastery at step t is the fraction of $\mathcal{H}$ with a trained visit by step t at which $k^{\text{true}} = G$, that is, on which all G responses were correct under the exact verifier. It is observed only on visits, so a prompt that the policy could solve but that has not been revisited since is not counted, and an arm that revisits prompts more often records mastery sooner. We therefore compare arms at equal numbers of steps, which are also equal numbers of rollouts for every arm except dynamic sampling, and read strict mastery together with held-out accuracy. Steps to 80% strict mastery is the first step at which 80% of $\mathcal{H}$ is mastered.

Error-prone set and its visit share. $\hat{\mathcal{E}}$ contains the 106 Math prompts that the judge over-credits, with $k^{\text{obs}} > k^{\text{true}}$, on at least half of their visits in the pooled uniform-sampling runs of seeds 0 and 1. Of these prompts, 90 lie in $\mathcal{H}$, and every prompt has 24 visits in the two runs. The set is fixed once and used for every arm. Because it is identified on uniform-sampling runs only, it is independent of every selector, replay and weighted run whose error-prone visit share it measures; for uniform sampling the share is fixed by design, since every prompt is visited equally. The error-prone visit share of a run is the fraction of its visits that land on $\hat{\mathcal{E}}$; it is $106/800 = 0.133$ under uniform sampling, which visits every prompt equally.

All-rejected solvable groups. An all-rejected solvable group is a group with $k^{\text{obs}} = 0$ and $k^{\text{true}} > 0$, in which the judge rejects every response although at least one is correct. The count is a lower bound on the groups that false negatives make look zero-variance, since a group whose judge verdicts are mixed can also contain rejected correct responses.

Throughput. Judge-informative groups per 1,000 rollouts is 1000 times the number of trained judge-informative groups divided by the number of rollouts generated. Each step generates 256 rollouts, or 384 for dynamic sampling. Rollouts per truly informative group is the inverse count for truly informative groups, over a stated window. For weighted arms, the Kish effective sample size (Kish, 1965) of step t is $(\sum_i w_{t,i})^2 / \sum_i w_{t,i}^2$ over the judge-informative groups of B_t; summed over steps and reported per 1,000 rollouts, it equals the number of judge-informative groups per 1,000 rollouts when all weights are equal. Visit concentration is the share of a run's visits received by its 10% most-visited prompts.

Held-out accuracy. MMLU-Pro STEM accuracy (Science) and accuracy on 500 level-balanced held-out MATH prompts (Math) are measured with greedy decoding at the saved checkpoints; the extraction rules and token limits are given in Appendix E.

Censoring rule. Four Math selector runs stopped before step 300, namely the first GRESO and MoPPS runs of seeds 0 and 1. The two GRESO runs stopped after 276 and 274 logged steps. The two MoPPS runs stopped after 279 and 288. For these runs every whole-run quantity (events, Λ_{vis} with its own T) is computed over the logged steps, and every window quantity uses the last fifth of the logged steps. A threshold that a run has not reached by its last logged step, such as 80% strict mastery, is right-censored at that step. It is reported as "$>$" followed by the step, it is left out of means, and ratios involving it are reported as ranges of the per-run values with the censored runs marked. Appendix J records why these runs stopped and that every rerun used the same configuration. Strict mastery and all-rejected solvable groups are further censored by visits, as

described above, so both are lower bounds, the first on the fraction of $\mathcal{H}$ the policy can solve and the second on the number of groups that false negatives make look zero-variance.

C JUDGES

This appendix describes the judges and how they were served (§C.1), the re-check protocol behind the measurements of response dependence (§C.2), the error rates of the judges we considered across modes and templates (§C.3), the calibration of the Science judges against a reference panel (§C.4), false negatives (§C.5), and the prompt and rubric templates verbatim (§C.6). None of the quantities here is a training run, so their uncertainties are item-level, namely a response-cluster bootstrap standard error (4,000 resamples of responses, or of groups where stated) for the Math judge, and a binomial standard error for the Science calibration, as indicated in each table.

C.1 JUDGES AND SERVING

Table 9 lists every model that produced a verdict used in the paper. The Math judge is reference-free. It sees the problem and the policy's full response, reasons for a bounded number of tokens and ends with a one-word verdict. The Science judges see the question, the reference answer that RaR-Science provides (Gunjal et al., 2026), the prompt's rubric and the full response, and return one Boolean per rubric item, from which the reward rule of Appendix B.2 computes the pass indicator. The training judges run on the training machine as two vLLM engines, each with tensor parallelism 2 and n-gram speculative decoding with 8 speculative tokens. They sample at temperature 0.6 and top-p 1.0 with at most 2,400 generated tokens. The Gemma 3 judge receives the prompt as raw text; the Gemma 4 judges receive it through their chat template, without a beginning-of-sequence token. The reference judge and the panel are called through a commercial inference API at temperature 0.6 and top-p 1.0 with at most 4,096 generated tokens. The auditing judges of the visit-triggered audit use the training judge's template of their domain at temperature 0.6 with at most 2,400 generated tokens. The Gemma-4-12B training judge was calibrated with the serving of its training runs. The stronger judge, Gemma-4-26B-A4B, was calibrated with a beginning-of-sequence token but served without one during training, and we did not repeat its calibration without the token; its agreement with the panel, and hence its position on the horizontal axis of Table 2c, describes the calibrated configuration, which may differ from the one that assigned the rewards.

Table 9: Judges, their roles and their templates. Thinking budget is the token budget stated in the prompt; every on-machine judge is capped at 2,400 generated tokens.

Model	Role	Template	Thinking budget	Serving
Gemma-3-12B-it (Gemma Team et al., 2025)	Math training judge	reference-free, reason then verdict	200 tokens	on machine, raw text
Gemma-4-12B-it (Gemma Team et al., 2026)	Science training judge	rubric, v1	400 tokens	on machine, chat template
Gemma-4-26B-A4B-it (25.2B total, 3.8B active)	stronger Science judge (judge-strength comparison)	rubric, v1	400 tokens	on machine, chat template
DeepSeek-V4-Flash (DeepSeek-AI, 2026)	reference judge; reward in the reference-as-reward control; auditing judge of the Science visit-triggered audit	rubric, v1	400 tokens	API
GLM-5.3-Flash (GLM-5 Team et al., 2026), reasoning effort high	calibration panel (rubric v2); auditing judge in one variant of the Math visit-triggered audit (Math thinking template)	rubric, v2	600 tokens	API
Kimi-K2.6 (Kimi Team et al., 2025), thinking off	calibration panel	rubric, v2	600 tokens	API

C.2 RE-CHECK PROTOCOL AND RESPONSE DEPENDENCE

Protocol. We sample base-policy responses to MATH prompts, grade them with the exact verifier, and keep 120 wrong responses, each to a different prompt, and 120 correct ones. Each response is sent to the judge ten times as an independent request, in the training mode (reason for at most 200 tokens, then a verdict) and with the training sampling parameters. For a wrong response y with A_y acceptances out of $K_{\mathrm{re}} = 10$ verdicts, we report three quantities. The base false-positive rate is $\sum_y A_y / \sum_y K_{\mathrm{re}}$. The *re-acceptance on re-check* is

$$\frac{\sum_y A_y(A_y - 1)}{\sum_y A_y(K_{\mathrm{re}} - 1)},$$

the probability that another verdict on the same response accepts it, given that one verdict did; it estimates $\mathbb{E}[a^2]/\mathbb{E}[a]$ over wrong responses, where a is a response's acceptance probability. The share of responses with at least one acceptance describes how widespread the errors are. We call the second quantity a re-acceptance, not an agreement. The probability that two verdicts on a wrong response agree, whether both accept or both reject, is $1 - 2(\mathbb{E}[a] - \mathbb{E}[a^2])$ and equals 0.858 ± 0.018 for the training mode, a number dominated by the many responses the judge rejects every time. The unweighted mean over responses of the per-response re-acceptance is 0.398 ± 0.050; it counts a response accepted once in ten like one accepted ten times, which is why we use the pooled estimator.

Same response versus another response. A high re-acceptance alone does not show that acceptance depends on the response, since per-prompt rates that vary across prompts, the model of noise-corrected GRPO (El Mansouri et al., 2026), also give $\mathbb{E}[\alpha^2]/\mathbb{E}[\alpha] > \mathbb{E}[\alpha]$. Under that model, however, a second verdict is equally likely to accept the same wrong response or another wrong response to the same prompt. We therefore measure the cross-response re-acceptance on every all-wrong group ($k^{\mathrm{true}} = 0$, with any number of acceptances) in the first 50 steps of the uniform-sampling Math runs of seeds 0 and 1, where the policy is still close to the base policy. In a group with k^{obs} acceptances among its eight wrong responses, the probability that another response is accepted given that one is equals $\sum k^{\mathrm{obs}}(k^{\mathrm{obs}} - 1) / \sum 7k^{\mathrm{obs}}$ over groups. Table 10 shows the ordering $0.68 > 0.49 > 0.22$, in which the same wrong response is re-accepted more often than another wrong response to the same prompt, which is in turn accepted more often than a random wrong response. The two rows, however, use different prompts selected in different ways. The re-checked responses come from 120 prompts, one each, while the all-wrong training groups come from harder prompts, whose base false-positive rate is 0.253 rather than 0.222. Under per-prompt rates each statistic estimates $\mathbb{E}[\alpha^2]/\mathbb{E}[\alpha]$ over its own mix of prompts, so the difference of $+0.19 \pm 0.07$ is suggestive rather than a test of response dependence; a test would re-ask the judge about each wrong response of the same groups. The evidence that acceptance depends on the response therefore rests on the training dynamics. Under response-independent errors a reward-hacking event requires the judge's rate on a prompt to rise while the verifier's falls, which Corollary 1 bounds by chance counts well below those measured. At the end of training (steps 250–299) the policy has converged on one wrong answer per prompt with a reward-hacking event, and the cross-response re-acceptance rises to the same-response level, which is consistent with the repeated acceptances across visits that Section 3 measures.

The first row is the main measurement of the paper (0.68 against a base rate of 0.22). The replicate through the serving stack of the training runs, with five times as many responses, gives a re-acceptance of 0.73 against a base rate of 0.21. Its same-minus-other difference is $+0.23 \pm 0.05$. The late false-positive rate of 0.455 is conditional on all-wrong groups, which concentrate on prompts with reward-hacking events; conditional on single wrong responses it is lower. Neither late number is a property of the judge alone, since both depend on which wrong answers the trained policy produces.

C.3 ERROR RATES ACROSS JUDGES, MODES AND TEMPLATES

Math. Before training we measured judges of three sizes in up to four modes on the same 240 responses (Table 11). In the thinking modes the judge reasons for at most 200 (medium) or 43 (low) tokens before its verdict; the bare mode asks for the verdict only; the commit-first mode first has the judge solve the problem and then compare the response's final answer with its own (Zhou, 2026). The flip rate is the unbiased pairwise disagreement between two verdicts on the same response. The

Table 10: Persistence of the Math judge's errors (Gemma-3-12B, training mode). The first two rows use 120 wrong and 120 correct base-policy responses with ten verdicts each (600 of each through the training serving stack in the second row) and a response-cluster bootstrap SE. The third row uses the all-wrong groups in steps 0–49 of uniform-sampling training (125 groups) and a group-cluster bootstrap SE. The fourth uses the all-wrong groups in steps 250–299 of the three uniform-sampling runs, as mean ± SE over three seeds.

Measurement	False-positive rate	Re-acceptance	Responses with at least one acceptance	False-negative rate
Same response, ten re-asks	0.222 ± 0.030	0.682 ± 0.049	58 of 120	0.163 ± 0.029
Same response, training serving stack	0.213 ± 0.014	0.727 ± 0.020	251 of 600	0.125 ± 0.012
Other wrong response, same prompt, steps 0–49	0.253 ± 0.024	0.494 ± 0.048	–	–
Other wrong response, same prompt, steps 250–299	0.455 ± 0.001	0.710 ± 0.014	–	–
Same minus other response, steps 0–49		$+0.19 \pm 0.07$		

mode governs the noise more than the size does. The 27B model in the medium thinking mode is no better than the 12B model. The verdict-first modes lower the false-positive rate but raise the false-negative rate and make verdicts nearly deterministic, with flip rates of 0.014 to 0.026. The 4B model accepts 73% of wrong responses. We trained with the 12B model in the medium thinking mode, which has the lowest balanced error of the four 12B modes. During training the same judge errs at different rates, because the responses differ. Its pooled false-positive rate in the uniform runs is 0.371 ± 0.005, against 0.222 on the calibration responses. Its per-response false-negative rate on groups the verifier accepts in full is 0.0058 ± 0.0001 (Appendix D), against 0.163. This is why the model of Section 2 lets α_i and β_i depend on θ.

Table 11: Math judges against the exact verifier on 120 wrong and 120 correct base-policy responses, ten verdicts each (response-cluster bootstrap SE). Balanced error is the mean of the two error rates; the flip rate is the pairwise disagreement between verdicts on the same response.

Judge	Mode	False-positive rate	False-negative rate	Balanced error	Flip rate	Re-acceptance
Gemma-3-4B	thinking, medium	0.731 ± 0.028	0.040 ± 0.011	0.386	0.140	0.845 ± 0.018
Gemma-3-12B	thinking, medium (training)	0.222 ± 0.030	0.163 ± 0.029	0.193	0.109	0.682 ± 0.049
Gemma-3-12B	thinking, low	0.264 ± 0.033	0.147 ± 0.027	0.205	0.109	0.736 ± 0.040
Gemma-3-12B	bare	0.188 ± 0.034	0.241 ± 0.038	0.214	0.026	0.906 ± 0.032
Gemma-3-12B	commit-first	0.178 ± 0.033	0.240 ± 0.038	0.209	0.023	0.924 ± 0.026
Gemma-3-27B	thinking, medium	0.212 ± 0.030	0.177 ± 0.030	0.195	0.100	0.712 ± 0.046
Gemma-3-27B	bare	0.079 ± 0.024	0.375 ± 0.043	0.227	0.017	–
Gemma-3-27B	commit-first	0.158 ± 0.032	0.195 ± 0.036	0.176	0.014	–

Science. Table 12 compares candidate rubric judges with the panel of §C.4 under the two rubric templates (both reproduced in §C.6). Template v2 asks for evidence before each decision and defaults to false. It raises Gemma-3-12B's binary pass agreement from 0.660 to 0.764. It lowers Gemma-4-12B's from 0.764 to 0.635, so each template helps one judge and hurts the other. Before training we fixed a band of 0.70 to 0.90 for the training judge's per-criterion agreement with the panel, so that the training judge errs often while its reward still tracks the panel; Gemma-4-12B with template v1, at 0.760, lies in this band. Gemma-4-26B-A4B, Gemma-4-31B, gpt-oss-120b and the DeepSeek reference agree with the panel on 91–93% of criteria, close to the panel's agreement with itself.

Table 12: Science rubric judges against the panel (190 RaR-Science prompts, eight base-policy completions each; criteria on which both panel members agree). False credit is $\Pr(\text{judge met} \mid \text{panel unmet})$ and false denial $\Pr(\text{judge unmet} \mid \text{panel met})$; binary pass agreement compares pass indicators per completion, computed with the score threshold and the Pitfall gate but without the Essential gate. Each judge is scored on 10,199 to 10,528 criteria. The binary agreement uses 1,472 to 1,520 completions. Binomial SE over criteria or completions; it ignores clustering within completions and prompts, so it understates the uncertainty.

Judge (template)	Agreement	False credit	False denial	Binary agreement	Pass rate, judge / panel
Gemma-3-12B (v1)	0.611 ± 0.005	0.418 ± 0.005	0.112 ± 0.010	0.660 ± 0.012	0.378 / 0.046
Gemma-3-12B (v2)	0.655 ± 0.005	0.368 ± 0.005	0.116 ± 0.010	0.764 ± 0.011	0.242 / 0.046
Gemma-4-12B (v1, training)	0.760 ± 0.004	0.242 ± 0.004	0.219 ± 0.014	0.764 ± 0.011	0.268 / 0.046
Gemma-4-12B (v2)	0.655 ± 0.005	0.372 ± 0.005	0.073 ± 0.008	0.635 ± 0.012	0.408 / 0.046
Gemma-4-26B-A4B (v1, stronger)	0.933 ± 0.002	0.064 ± 0.002	0.100 ± 0.010	0.941 ± 0.006	0.098 / 0.045
Gemma-4-31B (v1)	0.932 ± 0.002	0.064 ± 0.002	0.111 ± 0.010	0.947 ± 0.006	0.089 / 0.045
gpt-oss-120b (v1)	0.911 ± 0.003	0.087 ± 0.003	0.106 ± 0.009	0.937 ± 0.006	0.098 / 0.046
Qwen3-32B (v1)	0.731 ± 0.004	0.283 ± 0.005	0.124 ± 0.011	0.765 ± 0.011	0.276 / 0.045
DeepSeek-V4-Flash (v1, reference)	0.931 ± 0.002	0.056 ± 0.002	0.201 ± 0.013	0.964 ± 0.005	0.069 / 0.045

C.4 Calibration against the reference panel

The panel consists of GLM-5.3-Flash with high reasoning effort and Kimi-K2.6 without thinking, both prompted with template v2. We take as ground truth only the criteria on which the two agree; they agree on 0.935 ± 0.002 of 11,256 criteria (binomial SE), which indicates the agreement to expect between two strong judges. The calibration set is 190 RaR-Science prompts with eight base-policy completions each. A completion's panel pass indicator applies the reward rule to the panel's criterion verdicts, and a completion counts toward the binary rates below only if the two panel members give it the same pass indicator (1,471 completions, of which 2.9% pass).

Table 13 gives the binary error rates of the training rule. The training judge passes 21.3% of the completions the panel fails, and the reference judge 0.8%. The reference judge rarely repeats the training judge's false passes, of which it falsely passes 3%, so the two share a false pass on about 0.6% of the failing completions. A larger and different set is formed by the 11,301 base-policy completions of the unfiltered RaR-Science prompts that received a parsed verdict from every judge. On it the reward rule passes $38.8\% \pm 0.5\%$ of completions with Gemma-3-12B's verdicts, $30.3\% \pm 0.4\%$ with the training judge's and $6.9\% \pm 0.2\%$ with the reference judge's (binomial SE, ignoring prompt clustering).

Content-free responses. Responses that merely repeat the question pass the training judge $82.3\% \pm 1.2\%$ of the time (1,034 judgements of 88 such responses to held-out prompts; binomial SE), although the reference answer is in view. This rate is within the range that prior work reports for open-weight LLM judges used as reward models. Zhao et al. (2025) find that content-free "master key" responses, such as a lone colon or the opener "Thought process:", obtain false-positive rates as high as 80% from reference-based LLM judges; per-dataset averages reach 66% to 89% for Llama3-70B-Instruct, while frontier judges such as GPT-4o and Claude-4 stay far lower. For "Thought process:" the rate reaches 60% to 90% with Llama3-70B-Instruct and Qwen2.5-72B-Instruct. A trained generative verifier accepts a single blank space as a correct MATH answer 66.8% of the time. Zheng et al. (2025b) show more generally that a fixed response, optimized against the judge, can game an automatic benchmark. In rubric-based RL on science questions, Mahmoud et al. (2026) report that 63% to 75% of the criteria a GPT-4o-mini verifier newly credits during training are incorrect according to a reference panel. The restatement pass rate of our training judge is therefore a property of current practice rather than of an unusually weak judge, and the stronger judge of the judge-strength comparison passes far fewer failing responses (Table 12).

Table 13: Binary pass verdicts of the Science judges under the training reward rule against the panel's pass indicator (1,471 completions on which the panel members agree; binomial SE; the false-negative rate rests on about 43 panel passes).

Judge (template)	Agreement	False-positive rate	False-negative rate	Pass rate
Gemma-4-12B (v1, training)	0.789 ± 0.011	0.213 ± 0.011	0.163 ± 0.057	0.232 ± 0.011
Gemma-4-12B (v2)	0.704 ± 0.012	0.304 ± 0.012	0.023 ± 0.023	0.324 ± 0.012
Gemma-3-12B (v1)	0.708 ± 0.012	0.298 ± 0.012	0.093 ± 0.044	0.316 ± 0.012
DeepSeek-V4-Flash (v1, reference)	0.989 ± 0.003	0.008 ± 0.002	0.093 ± 0.044	0.035 ± 0.005

The judge-strength comparison (Table 2c) places each training judge by its binary pass agreement with the panel, which is 0.764 ± 0.011 for Gemma-4-12B, 0.941 ± 0.006 for Gemma-4-26B-A4B (measured with a beginning-of-sequence token; §C.1) and 0.964 ± 0.005 for the reference judge. These are the binary agreements of Table 12; Table 13 applies the full training rule, including the Essential gate, which is why the training judge's binary agreement is 0.764 in one table and 0.789 in the other.

C.5 False negatives

False negatives matter for selection in two ways. A prompt the policy has mastered keeps producing judge-informative groups when the judge rejects some correct responses (Proposition 1), so a selector does not retire it; and a solvable prompt can look unsolved when the judge rejects every correct response, which a streak-based selector then skips. The Math judge rejects 0.163 ± 0.029 of correct responses in the training mode and 0.24 in the verdict-first modes (Table 11). The Science training judge denies 0.219 ± 0.014 of criteria the panel marks as met, and fails 0.163 ± 0.057 of the completions the panel passes (Table 13); since a pass requires every Essential criterion, a single false denial on one of them is enough to fail a correct response. Appendix F reports how often training groups are all-rejected solvable groups, which the judge fails entirely although the verifier accepts at least one response.

C.6 Prompt and rubric templates

Text in angle brackets is replaced by the problem, the policy's full response, the thinking budget, the reference answer and the rubric. A verdict that cannot be parsed counts as a rejection (Math) or as all criteria unmet (Science).

Math judge, thinking modes (budget 200 in the training mode and for the auditing judge).

```
Problem:
<problem>
Student's answer:
<response>
Decide whether the student's final answer is correct.
Think it through in at most <budget> tokens, then end your
reply with exactly one word on its own line:  CORRECT or
INCORRECT.
```

The verdict is the last line consisting of CORRECT or INCORRECT, or failing that the last occurrence of either word.

Math judge, bare mode.

```
Problem:
<problem>
Student's answer:
<response>
```

```
Is the student's final answer correct?  Reply with exactly
one word:  CORRECT or INCORRECT.
```

Math judge, commit-first mode. The judge first solves each problem once with the prompt "Solve this problem. Give only the final answer on the last line, in the form Answer: $Answer." followed by the problem, and the solution is reused for every response to that problem in the following prompt.

```
Problem:
<problem>
Your own worked answer to this problem:
<judge's solution>
Student's answer:
<response>
Compare the student's FINAL ANSWER to yours.  Mathematically
equivalent forms count as the same answer.  Reply with
exactly one word:  CORRECT or INCORRECT.
```

In the verdict-first modes the first verdict line is used.

Science policy prompt. The policy receives each question after the instruction

```
Answer the following science question.  Give a complete,
well-organized response that states the answer and the
reasoning behind it.
```

Rubric judge, template v1 (training, stronger, reference and Science auditing judges; budget 400).

```
Question:
<question>
Reference answer (a guide, may be incomplete; the rubric is
authoritative):
<reference answer>
Rubric:
<one line per item:  "id.  [category, weight w] text">
Response to grade:
<response>
For each rubric item decide whether the response satisfies
it.  For a Pitfall item, true means the response AVOIDS the
pitfall.  Think it through in at most <budget> tokens, then
end your reply with exactly one JSON object on its own line
mapping every item number to true or false, for example
{"1":  true, "2":  false}.
```

Rubric judge, template v2 (panel; budget 600).

```
You are grading a response to a science question against
a rubric.  Be strict:  an item is satisfied ONLY if the
response explicitly contains it.  If the response is empty,
off-topic, nonsensical, cut off before saying anything
substantive, or merely restates the question, then NO item
is satisfied.
Question:
<question>
Reference answer (a guide, may be incomplete; the rubric is
authoritative):
<reference answer>
Response to grade:
<response>
Rubric:
```

```
<one line per item>
For EACH item, first write the item number and the shortest
quote from the response that satisfies it, or the word NONE
if nothing does; then decide.  For an item marked [Pitfall],
the text describes a MISTAKE to avoid:  write true only if
the response engages with the question AND does not make
that mistake; write false if it makes the mistake OR if the
response is empty, off-topic or nonsensical.  Default to
false whenever you are unsure.  Keep the whole reply under
<budget> tokens and end with exactly one JSON object on its
own line mapping every item number to true or false, for
example {"1":  false, "2":  true}.
```

The rubric items and their categories (Essential, Important, Optional, Pitfall) and weights come from RaR-Science (Gunjal et al., 2026); Pitfall weights are negative, and the weighted score of Appendix B.2 uses their absolute values, with "met" meaning that the pitfall is avoided. The verdicts are read from the last JSON object in the reply whose keys are item numbers; an item missing from it counts as unmet, and an empty response is shown to the judge as "(empty response)".

D Proofs and method details

D.1 Setting and conventions for Section 4

Fix a step, the parameters θ and a prompt x_i; we drop the index i where no confusion arises. The G responses $y_1, \dots, y_G$ are drawn i.i.d. from $\pi_\theta(\cdot \mid x_i)$, and $s(y) = \nabla_\theta \log \pi_\theta(y \mid x_i)$ is the sequence-level score, the sum of the token scores. We assume the score is bounded, $S = \sup_y \|s(y)\| < \infty$, which holds for a softmax policy over responses of bounded length; this licenses differentiation under the expectation and gives $\mathbb{E}[s(y)] = 0$. The analysis rests on four conditions.

(A1) Per-response verdicts. The verdict on response j is $r_j = J(x_i, y_j, \zeta_j) \in \{0, 1\}$, where the judge randomness $\zeta_1, \dots, \zeta_G$ is i.i.d. and independent of the responses, so that the acceptance probability $c(y) = \Pr(J = 1 \mid x_i, y)$ does not depend on θ. In the notation of Section 2, $c(y) = V(y)(1-b_i(y)) + (1-V(y))a_i(y)$, $p_i = \mathbb{E}[V(y)]$, $\alpha_i = \mathbb{E}[(1-V)a_i]/(1-p_i)$, $\beta_i = \mathbb{E}[Vb_i]/p_i$ and $q_i = \mathbb{E}[c(y)] = p_i(1-\beta_i) + (1-p_i)\alpha_i$, with all expectations over $y \sim \pi_\theta(\cdot \mid x_i)$. The false-positive rate $\alpha_i(\theta)$ and the false-negative rate $\beta_i(\theta)$ move with the policy unless a_i and b_i are constant over responses (response-independent error). A judge that grades a group jointly violates (A1).

(A2) Exchangeable group advantages. The advantage of response j is $f(r_j, K)$ with $K = \sum_j r_j$ and $\sum_j f(r_j, K) = 0$. GRPO's advantage as implemented in TRL is $f(r, K) = (r - K/G)/(\sigma_K + \epsilon)$ with the sample standard deviation $\sigma_K = \sqrt{K(G-K)/(G(G-1))}$ and $\epsilon = 10^{-4}$ (so $f \equiv 0$ when $K \in \{0, G\}$); the mean-only advantage of Dr. GRPO is $f(r, K) = r - K/G$.

(A3) One optimizer step per batch. Every run takes one optimizer step per generated batch, so the likelihood ratio equals one where the gradient is taken, the clip range never binds and the group gradient is $g = \sum_j f(r_j, K)\, s(y_j)$ up to the normalizer. Expectations are under the sampling policy; the sequence-level sampler-mismatch weight of Section 2, a factor close to one per response, is ignored.

(A4) Fixed normalizer. The token-level loss is divided by a normalizer Z_t fixed before the draw, as in the importance-weighted arms, and we absorb Z_t^{-1} into g. Arms that divide by the realized number of tokens deviate from this by a relative amount of order $1/P$, since one group contributes about a fraction $1/P$ of Z_t; we state this without a formal bound.

All statements concern expected gradients. Adam rescales coordinates after the gradient is formed, so the claims are first order in the gradient; where a parameter change is needed we allow any positive-definite preconditioner H held fixed within the step. The Science reward adds a soft over-long penalty to the rubric verdict and is not binary, so Lemma 1, Proposition 3 and Corollary 1 are stated for Math; Propositions 1 and 2 and Observation 1 use only (A1) and apply to both domains when informativeness is read from the verdicts.

D.2 PROPOSITION 1

ASSUME: (A1), and $y_1, \dots, y_G$ i.i.d. from $\pi_\theta(\cdot \mid x_i)$.
PROVE: $K \sim \mathrm{Bin}(G, q_i)$, so $\Pr(0 < K < G) = 1 - q_i^G - (1-q_i)^G$; this is $1-(1-\alpha_i)^G-\alpha_i^G$ if $p_i = 0$ and $1 - (1-\beta_i)^G - \beta_i^G$ if $p_i = 1$.

⟨1⟩1. The pairs (y_j, ζ_j), $j = 1, \dots, G$, are i.i.d. PROOF: The responses are i.i.d., the judge randomness is i.i.d., and the two are independent by (A1).

⟨1⟩2. The verdicts r_j are i.i.d. Bernoulli with mean q_i. PROOF: Each r_j is the same function of (y_j, ζ_j), so they are i.i.d. by ⟨1⟩1, and $\Pr(r_j = 1) = \mathbb{E}_y[c(y)] = q_i$.

⟨1⟩3. $K \sim \mathrm{Bin}(G, q_i)$. PROOF: K is a sum of G i.i.d. Bernoulli(q_i) variables by ⟨1⟩2.

⟨1⟩4. $\Pr(0 < K < G) = 1 - q_i^G - (1-q_i)^G$. PROOF: $\Pr(K = G) = q_i^G$ and $\Pr(K = 0) = (1-q_i)^G$ by ⟨1⟩3.

⟨1⟩5. $q_i = \alpha_i$ if $p_i = 0$ and $q_i = 1 - \beta_i$ if $p_i = 1$. PROOF: Substitute into $q_i = p_i(1-\beta_i) + (1-p_i)\alpha_i$.

⟨1⟩6. Q.E.D. PROOF: By ⟨1⟩4 and ⟨1⟩5.

For $G = 8$ the values are

$$1 - 0.8^8 - 0.2^8 = 0.83223, \qquad 1 - 0.7^8 - 0.3^8 = 0.94229,$$
$$1 - 2 \cdot 0.5^8 = 0.99219, \qquad 1 - 0.9942^8 - 0.0058^8 = 0.0455.$$

Whether a_i depends on the response plays no role at a single visit, since only the marginal rate q_i enters; it matters for the dynamics (Proposition 3).

Heterogeneous rates lower the average.

ASSUME: $h(\alpha) = 1 - (1-\alpha)^G - \alpha^G$ and prompt rates α_i with mean $\bar{\alpha}$.
PROVE: the mean of $h(\alpha_i)$ is at most $h(\bar{\alpha})$.

⟨1⟩1. $h''(\alpha) = -G(G-1)\big[(1-\alpha)^{G-2} + \alpha^{G-2}\big] < 0$ on $[0, 1]$. PROOF: Differentiate twice.

⟨1⟩2. Q.E.D. PROOF: h is concave by ⟨1⟩1; apply Jensen's inequality.

In the uniform Math runs, groups in which the verifier accepts none of the eight responses are judge-informative 0.645 ± 0.004 of the time. This is below $h(0.22) = 0.863$ at the base false-positive rate of the re-check measurement (Appendix C). It is also below $h(0.371) = 0.975$ at the runs' pooled in-training false-positive rate, as the concavity implies. A Beta$(0.59, 1.74)$ distribution of α_i reproduces a judge-informative probability of 0.640. Groups the verifier accepts in full are judge-informative only 0.036 ± 0.001 of the time. The Math judge's per-response false-negative rate on these groups is 0.0058 ± 0.0001. At that rate the formula gives 0.046 ± 0.001 when evaluated per seed, and 0.045 at the pooled rate (the value marked in Figure 4a). It differs from the false-negative rate of 0.163 ± 0.029 in Appendix C because the two measurements concern different correct responses, since the re-check sample holds base-policy responses to the prompts of the re-check protocol, whereas the in-training rate is measured on prompts the policy already solves in all eight attempts; the second rate decides whether a learned prompt stays informative. Prompts the policy never solves are thus judge-informative 18.0 ± 0.4 times as often as prompts it solves in full.

Precedent and effect on the selectors. Yu (2026) describe the one-step version of this mechanism for shaped rewards, where all-fail groups keep a nonzero spread and pass zero-variance filters; Proposition 1 is its counterpart for judge errors on binary verdicts. Under the exact verifier, GRESO and MoPPS reach visit amplifications of 6.62 ± 0.22 and 9.97 ± 0.27 in Math. Under the judge they reach 11.40 ± 0.10 and 13.18 ± 0.09. The ratios, computed per seed, are 1.73 ± 0.04 and 1.32 ± 0.03. The judge raises amplification by lengthening the time a prompt stays informative. Figure 4(a) plots the probability of an informative group for $G = 8$ and $G = 16$.

Style-driven splits. In the notation of (A1), a split driven by style is one in which the acceptance probability $c(y)$ varies with surface features of the response, such as restating the question or giving a terse final answer, that are independent of correctness. Such an error is response-dependent in our sense, since a_i and b_i vary over responses, but it does not resolve as the policy learns the prompt.

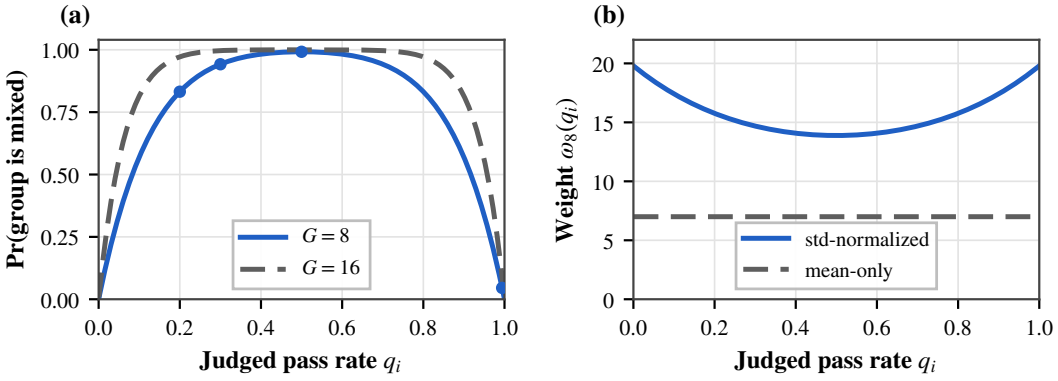


Figure 4: Closed-form curves for Proposition 1 and Lemma 1. **(a)** Probability that a group of G verdicts is informative as a function of the judged pass rate q_i; the markers are the values of Proposition 1 for $G = 8$ (Appendix D.2). A prompt the policy never solves, with a false-positive rate of 0.2, is almost as likely to look informative as a prompt with $q_i = 1/2$, and larger groups make this worse. **(b)** The weight $\omega_8(q_i)$ that turns ∇q_i into the expected group gradient, for GRPO's standard-deviation-normalized advantage (minimum 13.88 at $q_i = 1/2$) and for the mean-only advantage (constant $G - 1 = 7$). The weight never vanishes, so every informative prompt passes its judged pass-rate gradient into the update.

When the policy never or always solves the prompt, $q_i = \alpha_i$ or $q_i = 1 - \beta_i$, and a judge that splits on style keeps these rates away from 0 and 1, so by Proposition 1 the prompt stays informative; a split driven by correctness has α_i and β_i near zero and becomes zero-variance as p_i approaches 0 or 1. Persistently split prompts are therefore enriched for style-driven splits, a selector that keeps them changes the composition of its batches, and by Proposition 3 the term C_i moves the policy toward the style the judge accepts. This composition effect does not require any single prompt to be visited more often, which is why it can act before visit amplification grows (Section 3).

D.3 Proposition 2

The selector's step is $g_t = P^{-1} \sum_{i \in B_t} w_{t,i} g_{t,i}$ with inclusion probability $\rho_{t,i}$, weight $w_{t,i}$ and weighted exposure $X_{t,i} = \rho_{t,i} w_{t,i}$. Write $\bar{g}_{t,i} = \mathbb{E}[g_{t,i} \mid \mathcal{F}_t]$ for the mean group gradient of prompt i at θ_t, and $b_{t,i} = \mathbb{E}[g^J_{t,i} - g^V_{t,i} \mid \mathcal{F}_t]$ for the bias of judge rewards against exact rewards on the same responses.

> Assume: 1. B_t and $w_{t,i}$ are chosen from $\mathcal{F}_t$, possibly with extra randomness independent of the rollouts, before the groups of B_t are rolled out with fresh randomness. 2. $\|b_{t,i}\| \le \Delta$ for i in a set $\mathcal{E}$ with $|\mathcal{E}| \le m$, and $b_{t,i} = 0$ otherwise.
>
> Prove: $\mathbb{E}[g_t \mid \mathcal{F}_t] = P^{-1} \sum_i X_{t,i} \bar{g}_{t,i}$, and $\| \sum_{t \le T} \mathbb{E}[g^J_t - g^V_t \mid \mathcal{F}_t] \| \le \Delta m \Lambda T / N$, with equality for $m = 1$ when the exposures do not depend on $\mathcal{E}$ and $b_{t,i^*} = \Delta u$ for a fixed unit vector u on the most exposed prompt i^*.

⟨1⟩1. Given $\mathcal{F}_t$ and $\{i \in B_t\}$, $g_{t,i}$ has mean $\bar{g}_{t,i}$. Proof: By assumption 1, the rollouts of i do not depend on the inclusion decision.

⟨1⟩2. $\mathbb{E}[g_t \mid \mathcal{F}_t] = P^{-1} \sum_i \rho_{t,i} w_{t,i} \bar{g}_{t,i} = P^{-1} \sum_i X_{t,i} \bar{g}_{t,i}$. Proof: Write $g_t = P^{-1} \sum_i \mathbf{1}\{i \in B_t\} w_{t,i} g_{t,i}$, use that $w_{t,i}$ is $\mathcal{F}_t$-measurable, and apply ⟨1⟩1.

⟨1⟩3. The selector's expected step is uniform sampling's with prompt i reweighted by $N X_{t,i}/P$. Proof: Uniform sampling ($\rho_{t,i} = P/N$, $w_{t,i} = 1$) has $X_{t,i} = P/N$ and expected step $N^{-1} \sum_i \bar{g}_{t,i}$ by ⟨1⟩2.

⟨1⟩4. $\mathbb{E}[g^J_t - g^V_t \mid \mathcal{F}_t] = P^{-1} \sum_{i \in \mathcal{E}} X_{t,i} b_{t,i}$. Proof: Apply ⟨1⟩2 to the difference of the two rewards on the same responses; $b_{t,i} = 0$ outside $\mathcal{E}$.

⟨1⟩5. $\| \sum_{t \le T} \mathbb{E}[g^J_t - g^V_t \mid \mathcal{F}_t] \| \le (\Delta/P) \sum_{i \in \mathcal{E}} \sum_{t \le T} X_{t,i}$. Proof: Triangle inequality with $X_{t,i} \ge 0$ and $\|b_{t,i}\| \le \Delta$.

⟨1⟩6. $\sum_{i\in\mathcal{E}}\sum_{t\le T} X_{t,i} \le m \max_i \sum_{t\le T} X_{t,i} = m\Lambda TP/N$. PROOF: $|\mathcal{E}| \le m$ and the definition $\Lambda = \max_i \sum_t X_{t,i}/(TP/N)$.

⟨1⟩7. The bound is $\Delta m\Lambda T/N$. PROOF: Combine ⟨1⟩5 and ⟨1⟩6. It holds pathwise, since it uses only the realized cumulative exposures, which may depend on $\mathcal{E}$.

⟨1⟩8. Under the equality conditions the bound is attained. PROOF: The sum in ⟨1⟩5 is $P^{-1}\sum_t X_{t,i^*}\Delta u$, a nonnegative multiple of u with norm $\Delta\Lambda T/N$; if the exposures depended on $\mathcal{E}$, i^* would depend on the error and attainment would not be guaranteed.

⟨1⟩9. Q.E.D. PROOF: By ⟨1⟩2, ⟨1⟩3, ⟨1⟩7 and ⟨1⟩8.

Sparse and dense errors. Let Λ_m be the mean of the m largest cumulative exposures divided by TP/N, so that $\Lambda_1 = \Lambda$.

ASSUME: the exposures sum to P at every step, as for every design that draws P prompts with unit weights and for the importance-weighted designs ($X_{t,i} = P/N$).

PROVE: the bound of Proposition 2 sharpens to $\Delta T \min(m\Lambda_m, N)/N$.

⟨1⟩1. $\sum_{i\in\mathcal{E}}\sum_t X_{t,i} \le m\Lambda_m TP/N$. PROOF: The m terms are at most the m largest cumulative exposures.

⟨1⟩2. $\sum_{i\in\mathcal{E}}\sum_t X_{t,i} \le \sum_i \sum_t X_{t,i} = TP$. PROOF: Exposures are nonnegative and sum to P per step.

⟨1⟩3. Q.E.D. PROOF: Insert the smaller of ⟨1⟩1 and ⟨1⟩2 into ⟨1⟩5 of the proof above.

For $m = N$ every such design has the bound ΔT, so the factor Λ matters only when the bias is concentrated on $m \ll N/\Lambda$ prompts. Response-independent error with a 22% false-positive rate biases every prompt with wrong responses, but by Proposition 3 that bias lies along the true direction; the sparse, misdirected part is the response-dependent term C_i. The measured top-m visit amplifications in Math are below (mean ± SE over three seeds).

	$m = 1$	$m = 10$	$m = 20$	$m = 50$
GRESO	11.40 ± 0.10	10.40 ± 0.09	9.28 ± 0.16	6.58 ± 0.07
MoPPS	13.18 ± 0.09	11.23 ± 0.04	10.01 ± 0.10	7.46 ± 0.01

The visit cap of 2.1 is checked before a visit is added, so its realized maximum is 26 visits. That is 2.17 times uniform sampling's 12 visits, for every m.

Exposure, visits and exactness. For an unweighted selector ($w_{t,i} = 1$) the exposure $X_{t,i} = \rho_{t,i}$ is the compensator of the visit count, so $n_{T,i} - \sum_{t\le T}\rho_{t,i}$ is a martingale and realized visits estimate cumulative exposure; the visit amplification Λ_{vis} reported in the paper is this estimate, and the maximum of noisy counts is biased upward, which is why an i.i.d. uniform draw would show a visit amplification above one. The factor Λ is a supremum ratio of cumulative exposures, analogous to, but not the same as, a concentrability coefficient, which compares occupancy measures (Munos, 2003; Chen & Jiang, 2019; Cortes et al., 2010). For the importance-weighted arms $X_{t,i} = \rho_{t,i} \cdot P/(N\rho_{t,i}) = P/N$, so $\Lambda = 1$ exactly (Proposition 4), and these arms fix the normalizer before the draw, so (A4) holds exactly. Dynamic sampling as implemented decides inclusion after seeing the group, which violates assumption 1.

ASSUME: dynamic sampling rolls $1.5P$ uniformly drawn prompts and keeps every informative group, which is the case whenever at most P of them are informative.

PROVE: its expected step is uniform sampling's times a global factor.

⟨1⟩1. Unanimous groups have zero gradient. PROOF: $f \equiv 0$ when $K \in \{0, G\}$ by (A2).

⟨1⟩2. The step equals P^{-1} times the sum of the group gradients of all $1.5P$ rolled prompts. PROOF: Every informative group is kept, and the dropped or padded groups contribute zero by ⟨1⟩1.

⟨1⟩3. Q.E.D. PROOF: The rolled prompts are drawn uniformly, so ⟨1⟩2 of the proof above applies with $X_{t,i} = 1.5P/N$.

When more than P groups are informative, the excess groups are dropped and the statement holds approximately; the same argument covers a selector that decides from $\mathcal{F}_t$ before rollout and also discards unanimous groups afterwards.

What the bound does not say. Proposition 2 bounds a worst case and does not determine harm. The capped arms have gaps of 0.071 to 0.086, about twice uniform sampling's 0.040, which the bound permits. The decisive comparison is the difficulty-matched replay, which has GRESO's visit amplification of 11.47 but a gap of 0.043 ± 0.002. Uniform sampling's gap is 0.040 ± 0.002, and the replayed schedule's is 0.117 ± 0.007 at the same amplification (Table 1); at equal exposure, which prompts are repeated therefore decides the harm. In Science, uniform sampling's share by step 50 is 3.94 visits. GRESO's most visited prompt has 7 visits in every seed, a visit amplification of 1.78 ± 0.00. MoPPS's reaches 6.09 ± 0.14 without a loss at that step, while GRESO trails uniform sampling by 4.3 ± 1.1 points. Uniform sampling's epoch schedule gives 5 visits (1.27), since the last partial batch of each epoch is dropped, and an i.i.d. uniform draw would give 2.67 on average. Dynamic sampling acquires its most frequent prompt 7 times by step 50 ($\Lambda_{\text{vis}} = 1.18$ with $P_r = 48$). By step 50 GRESO has thus exposed no prompt more than dynamic sampling has, and dynamic sampling shows no loss, so amplification does not explain GRESO's early deficit.

D.4 The finite-group identity

Proposition 3 rests on the following identity, which Davis & Recht (2025) prove for exact binary rewards (Mroueh (2025) give the version with population mean and standard deviation). The only new step is that a judge's verdicts may replace the exact rewards when they are per-response in the sense of (A1).

Lemma 1. *Under (A1)–(A4),* $\mathbb{E}[g_i] = \omega_G(q_i)\nabla_\theta q_i$ *with*

$$\omega_G(q) = \frac{\mathbb{E}_{K\sim\mathrm{Bin}(G,q)}[Kf(1,K)]}{q(1-q)} = G(G-1)\,\mathbb{E}_{J\sim\mathrm{Bin}(G-2,q)}\Big[\frac{f(1,J+1)}{G-1-J}\Big].$$

If $f(1,K) > 0$ *for* $0 < K < G$*, then* $\omega_{\min} = \inf_{q\in(0,1)}\omega_G(q) > 0$*. For GRPO's advantage,* $\omega_G(q) = (G-1)\,\mathbb{E}_{J\sim\mathrm{Bin}(G-2,q)}[1/(\sigma_{J+1}+\epsilon)]$*, so* $(G-1)/(\sigma_{\lfloor G/2\rfloor}+\epsilon) \le \omega_G \le (G-1)/(G^{-1/2}+\epsilon)$*; for* $G = 8$ *this gives* $13.09 \le \omega_8 \le 19.79$*.*

ASSUME: (A1)–(A4). Let $K' = \sum_{j\ge 2} r_j$ and $\varphi(r) = \mathbb{E}[f(r, r+K')]$ for $r \in \{0,1\}$.

PROVE: the statements of Lemma 1.

⟨1⟩1. $\mathbb{E}[g] = G\,\mathbb{E}[f(r_1,K)s(y_1)]$. PROOF: $g = \sum_j f(r_j,K)s(y_j)$ and the pairs (y_j,ζ_j) are exchangeable.

⟨1⟩2. $K' \sim \mathrm{Bin}(G-1,q)$ is independent of (y_1,ζ_1). PROOF: By ⟨1⟩1–⟨1⟩2 of the proof of Proposition 1; this is the only place where per-response verdicts are used.

⟨1⟩3. $\mathbb{E}[f(r_1,K) \mid y_1,\zeta_1] = \varphi(r_1)$. PROOF: $K = r_1 + K'$, r_1 is a function of (y_1,ζ_1), and ⟨1⟩2.

⟨1⟩4. $\mathbb{E}[g] = G\big(\varphi(1)-\varphi(0)\big)\mathbb{E}[r_1 s(y_1)]$. PROOF: By ⟨1⟩1 and ⟨1⟩3, $\mathbb{E}[g] = G\,\mathbb{E}[\varphi(r_1)s(y_1)]$; write $\varphi(r_1) = \varphi(0) + r_1(\varphi(1)-\varphi(0))$ and use $\mathbb{E}[s] = 0$.

⟨1⟩5. $\mathbb{E}[r_1 s(y_1)] = \nabla_\theta q$. PROOF: $\mathbb{E}[r_1 \mid y_1] = c(y_1)$, and $\mathbb{E}[c(y)s(y)] = \nabla_\theta \mathbb{E}_{\pi_\theta}[c(y)]$ because c does not depend on θ (A1).

⟨1⟩6. $G(\varphi(1)-\varphi(0)) = \mathbb{E}[Kf(1,K)]/(q(1-q))$ with $K \sim \mathrm{Bin}(G,q)$.

⟨2⟩1. $\mathbb{E}[K\,h(K)] = Gq\,\mathbb{E}[h(1+K')]$ and $\mathbb{E}[(G-K)h(K)] = G(1-q)\,\mathbb{E}[h(K')]$ for any h. PROOF: $K\,h(K) = \sum_j r_j h(K)$ and $(G-K)h(K) = \sum_j (1-r_j)h(K)$; condition on r_1 and use exchangeability.

⟨2⟩2. $G\varphi(1) = \mathbb{E}[Kf(1,K)]/q$. PROOF: First identity of ⟨2⟩1 with $h(K) = f(1,K)$.

⟨2⟩3. $G\varphi(0) = \mathbb{E}[(G-K)f(0,K)]/(1-q)$. PROOF: Second identity of ⟨2⟩1 with $h(K) = f(0,K)$.

⟨2⟩4. $(G-K)f(0,K) = -Kf(1,K)$. PROOF: The zero-sum condition of (A2) with K accepted responses.

⟨2⟩5. Q.E.D. PROOF: Subtract ⟨2⟩3 from ⟨2⟩2, use ⟨2⟩4, and note $1/q + 1/(1-q) = 1/(q(1-q))$.

⟨1⟩7. The second form of ω_G holds.

⟨2⟩1. $f(1,G) = 0$. PROOF: The zero-sum condition at $K = G$ reads $Gf(1,G) = 0$.

⟨2⟩2. $\binom{G}{K}K(G-K) = G(G-1)\binom{G-2}{K-1}$ for $1 \le K \le G-1$. PROOF: Expand the binomial coefficients.

⟨2⟩3. Q.E.D. PROOF: By ⟨2⟩1 only $1 \le K \le G-1$ contribute to $\mathbb{E}[Kf(1,K)]$; write $Kf(1,K) = K(G-K)\,f(1,K)/(G-K)$, apply ⟨2⟩2, divide by $q(1-q)$ and set $J = K-1$.

⟨1⟩8. If $f(1,K) > 0$ for $0 < K < G$, then $\omega_{\min} > 0$. PROOF: By ⟨1⟩7, $\omega_G(q)$ is a binomial average of the $G-1$ positive numbers $G(G-1)f(1,J+1)/(G-1-J)$ and is at least their minimum for every q.

⟨1⟩9. For GRPO, $\omega_G(q) = (G-1)\,\mathbb{E}[1/(\sigma_{J+1}+\epsilon)]$ with the stated bounds. PROOF: $f(1,K)/(G-K) = 1/(G(\sigma_K+\epsilon))$, and for $1 \le K \le G-1$, σ_K lies between $\sigma_1 = G^{-1/2}$ and $\sigma_{\lfloor G/2 \rfloor}$. For $G = 8$ the bounds are $7/(\sigma_4+\epsilon) = 13.09$ and $7/(\sigma_1+\epsilon) = 19.79$.

⟨1⟩10. Q.E.D. PROOF: By ⟨1⟩4–⟨1⟩9.

For GRPO's advantage with $G = 8$ and $\epsilon = 10^{-4}$, the weight is symmetric about $q = 1/2$ (Figure 4b) and takes the values

q	$\to 0$	0.1	0.2	0.3	0.5
$\omega_8(q)$	19.79	17.41	15.76	14.69	13.88

The minimum 13.88 at $q = 1/2$ is found by evaluating the polynomial, and the proofs below use only $\omega_{\min} \ge 13.09$. For the mean-only advantage, $\mathbb{E}[K(1-K/G)] = (G-1)q(1-q)$, so $\omega_G \equiv G-1$. For large G, $\omega_G(q) \approx \sqrt{G(G-1)}/\sqrt{q(1-q)}$, which is proportional to the derivative of $2\arcsin\sqrt{q}$, the transformation that Davis & Recht (2025) associate with GRPO.

D.5 PROPOSITION 3

Let $e = q - p$ be the judged-minus-true gap and $C = \mathbb{E}[(a-\alpha)(1-V)s] - \mathbb{E}[(b-\beta)Vs]$, with α and β held at their current values. Prompt i's own contribution to a step is $\eta H g$, with the weight $w_{t,i}/P$ absorbed into η and H positive definite.

ASSUME: (A1)–(A4), and e has a bounded Hessian in θ (true for a softmax policy over finitely many responses).

PROVE: (i) $\nabla q = (1-\alpha-\beta)\nabla p + C$ and $C = (1-p)\nabla\alpha - p\nabla\beta$. (ii) Under response-independent error, $C = 0$ and $\mathbb{E}[g] = \omega_G(q)(1-\alpha-\beta)\nabla p$, with $\|\nabla p\| \le pS$. (iii) The expected first-order change of e caused by the step is $\eta\,\omega_G(q)\langle \nabla e, H\nabla q\rangle$, which is at most zero under response-independent error with $\alpha+\beta \le 1$ and equals $\eta\,\omega_G(q)(C^\top HC + R)$ with $|R| \le \|H\|\,pS(pS + \|C\|)$ in general.

⟨1⟩1. $\nabla q = \mathbb{E}[Vs] - \mathbb{E}[Vb\,s] + \mathbb{E}[(1-V)a\,s]$. PROOF: $c = V - Vb + (1-V)a$, and $\nabla q = \mathbb{E}[c\,s]$ because c does not depend on θ.

⟨1⟩2. $\mathbb{E}[Vs] = \nabla p$ and $\mathbb{E}[(1-V)s] = -\nabla p$. PROOF: The first as in ⟨1⟩1 with c replaced by V; the second because $\mathbb{E}[s] = 0$.

⟨1⟩3. $\mathbb{E}[(1-V)a\,s] = -\alpha\nabla p + \mathbb{E}[(1-V)(a-\alpha)s]$. PROOF: Add and subtract $\alpha\,\mathbb{E}[(1-V)s]$ and use ⟨1⟩2.

⟨1⟩4. $\mathbb{E}[Vb\,s] = \beta\nabla p + \mathbb{E}[V(b-\beta)s]$. PROOF: Add and subtract $\beta\,\mathbb{E}[Vs]$ and use ⟨1⟩2.

⟨1⟩5. $\nabla q = (1-\alpha-\beta)\nabla p + C$. PROOF: Insert ⟨1⟩2–⟨1⟩4 into ⟨1⟩1.

⟨1⟩6. $C = (1-p)\nabla\alpha - p\nabla\beta$.

⟨2⟩1. $(1-p)\nabla\alpha = \mathbb{E}[(1-V)(a-\alpha)s]$. PROOF: Differentiate $(1-p)\alpha = \mathbb{E}[(1-V)a]$ with a fixed, giving $(1-p)\nabla\alpha - \alpha\nabla p = \mathbb{E}[(1-V)a\,s]$, and apply ⟨1⟩3.

⟨2⟩2. $p\nabla\beta = \mathbb{E}[V(b-\beta)s]$. PROOF: Differentiate $p\beta = \mathbb{E}[Vb]$, giving $p\nabla\beta + \beta\nabla p = \mathbb{E}[Vb\,s]$, and apply ⟨1⟩4.

⟨2⟩3. Q.E.D. PROOF: Subtract ⟨2⟩2 from ⟨2⟩1.

⟨1⟩7. Under response-independent error $C = 0$ and $\mathbb{E}[g] = \omega_G(q)(1-\alpha-\beta)\nabla p$. PROOF: $a-\alpha$ and $b-\beta$ vanish identically, so $C = 0$; apply Lemma 1 and ⟨1⟩5.

⟨1⟩8. $\|\nabla p\| \le pS$. PROOF: $\|\mathbb{E}[Vs]\| \le \mathbb{E}[V\|s\|] \le pS$.

⟨1⟩9. The expected first-order change of e is $\eta\,\omega_G(q)\langle\nabla e, H\nabla q\rangle + O(\eta^2)$. PROOF: Taylor's theorem gives $e(\theta+\eta Hg) - e(\theta) = \eta\langle\nabla e, Hg\rangle + O(\eta^2\|g\|^2)$; take the expectation with Lemma 1, and note that $\mathbb{E}\|g\|^2$ is bounded because s and f are.

⟨1⟩10. $\langle \nabla e, H\nabla q\rangle = -(\alpha+\beta)(1-\alpha-\beta)\nabla p^\top H\nabla p + \big(1-2(\alpha+\beta)\big)\nabla p^\top HC + C^\top HC$. PROOF: $\nabla e = \nabla q - \nabla p = -(\alpha+\beta)\nabla p + C$ by ⟨1⟩5; expand with H symmetric.

⟨1⟩11. Under response-independent error with $\alpha + \beta \le 1$ the change is at most zero. PROOF: By ⟨1⟩7 only the first term of ⟨1⟩10 remains, and it is at most zero because H is positive definite.

⟨1⟩12. The first two terms of ⟨1⟩10 satisfy $|R| \le \|H\|\, pS(pS + \|C\|)$. PROOF: $(\alpha+\beta)(1-\alpha-\beta) \le 1/4$ and $|1 - 2(\alpha+\beta)| \le 1$ for $\alpha + \beta \in [0, 1]$, and ⟨1⟩8.

⟨1⟩13. Q.E.D. PROOF: (i) by ⟨1⟩5 and ⟨1⟩6; (ii) by ⟨1⟩7 and ⟨1⟩8; (iii) by ⟨1⟩9–⟨1⟩12.

The exact-reward update is $\omega_G(p)\nabla p$ by Lemma 1 with $c = V$, so under response-independent error the judged update is the true direction scaled by $(1 - \alpha - \beta)\,\omega_G(q)/\omega_G(p)$; for GRPO at $G = 8$ the ratio of weights lies in $[0.70, 1.43]$ with the evaluated minimum. With the proven bound it lies in $[0.66, 1.51]$. The factor $1 - \alpha - \beta$ is Youden's index in the analysis of Rad et al. (2026) and the correction factor of noise-corrected GRPO (El Mansouri et al., 2026), both of which assume response-independent errors. Cai et al. (2025, App. B) derive the covariance term C as the residual bias of their correction when the verifier's errors depend on the response. A prompt the policy never solves is informative with probability $1-(1-\alpha)^G-\alpha^G$ (Proposition 1) yet has an expected update of order p under response-independent error, so it attracts compute without teaching, much as a noisy TV attracts a curiosity-driven agent (Burda et al., 2019a); a response-dependent error is learnable, so a learning-progress criterion registers progress on it.

Summed over visits.

ASSUME: at each of the prompt's visits $k = 1, \dots, n$ the preconditioner has smallest eigenvalue at least $\lambda > 0$ and norm at most $\|H\|$, $\|C_k\| \ge \kappa$, $\|C_k\| \le C_{\max}$ and $p_k \le \bar{p}$, with $B = \lambda\kappa^2 - \|H\|\bar{p}S(\bar{p}S + C_{\max}) > 0$.

PROVE: the first-order increase of the gap caused by the prompt's own updates is at least $\eta\,\omega_{\min} Bn$.

⟨1⟩1. At visit k the increase is $\eta\,\omega_G(q_k)(C_k^\top H_k C_k + R_k)$. PROOF: Proposition 3(iii).

⟨1⟩2. $C_k^\top H_k C_k + R_k \ge B$. PROOF: $C_k^\top H_k C_k \ge \lambda\kappa^2$, and $|R_k|$ is bounded by (iii) with $p_k \le \bar{p}$ and $\|C_k\| \le C_{\max}$.

⟨1⟩3. $\omega_G(q_k) \ge \omega_{\min}$. PROOF: Lemma 1.

⟨1⟩4. Q.E.D. PROOF: Multiply ⟨1⟩2 and ⟨1⟩3, which are both positive, and sum over the n visits.

The condition $\|C_k\| \ge \kappa$ holds only while the policy can still change its false-positive rate, that is, while the judge accepts some of the policy's wrong responses more often than others; it fails once a is constant on their support. This is what Section 4 means by a response-dependent error growing with visits, and it is consistent with the gap measured by visit number in Figure 2b. The contrast agrees with findings that the pattern of verifier errors, not their rate, determines their effect (Egashira et al., 2026) and that a wrong label harms training once the policy can produce the answer it rewards (Yang et al., 2026c); under plain GRPO, where each prompt receives few visits, response-dependent verifier errors need not grow (Zhang, 2026).

Components orthogonal to the true direction. A complementary statement concerns summed updates rather than the gap. Let $\mathcal{F}_k$ be the history before the rollouts of visit k, g_k its group gradient, $\omega_k = \omega_G(q_k)$, $\xi_k = g_k - \mathbb{E}[g_k \mid \mathcal{F}_k]$ and Π_k the projection orthogonal to ∇p_k, and $S_n = \sum_{k \le n} \Pi_k g_k$.

ASSUME: $\mathbb{E}[\|\xi_k\|^2 \mid \mathcal{F}_k] \le \sigma^2$ and a fixed number of visits n (or, by optional stopping, a bounded stopping time).

PROVE: under response-independent error $\mathbb{E}\|S_n\|^2 \le n\sigma^2$; if instead $\langle \Pi_k C_k, u\rangle \ge \gamma > 0$ at every visit for a fixed unit vector u, then $\langle \mathbb{E}S_n, u\rangle \ge \gamma\,\omega_{\min} n$.

⟨1⟩1. $\Pi_k g_k = \omega_k \Pi_k C_k + \Pi_k \xi_k$. PROOF: $\mathbb{E}[g_k \mid \mathcal{F}_k] = \omega_k[(1 - \alpha_k - \beta_k)\nabla p_k + C_k]$ by Lemma 1 and (i), and $\Pi_k \nabla p_k = 0$.

⟨1⟩2. Under response-independent error $S_n = \sum_{k \le n} \Pi_k \xi_k$. PROOF: $C_k = 0$ in ⟨1⟩1.

⟨1⟩3. $\mathbb{E}\langle \Pi_k \xi_k, \Pi_l \xi_l\rangle = 0$ for $k < l$. PROOF: $\Pi_k \xi_k$ is $\mathcal{F}_l$-measurable and $\mathbb{E}[\Pi_l \xi_l \mid \mathcal{F}_l] = 0$.

⟨1⟩4. $\mathbb{E}\|S_n\|^2 \le n\sigma^2$. PROOF: By ⟨1⟩2 and ⟨1⟩3, $\mathbb{E}\|S_n\|^2 = \sum_k \mathbb{E}\|\Pi_k \xi_k\|^2 \le \sum_k \mathbb{E}\|\xi_k\|^2$.

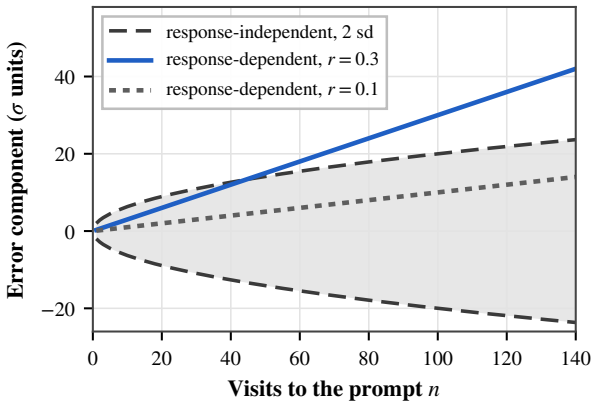


Figure 5: The summed update component orthogonal to the true direction (Appendix D.5) as a function of the number of visits to a prompt, in units of the per-visit noise σ. A response-dependent error with drift ratio $r = \gamma\omega_{\min}/\sigma$ grows linearly, while under response-independent error the component stays within a bound on its root-mean-square size that grows as $\sqrt{n}$ (shaded, twice the bound). With $r = 0.3$ the drift leaves the envelope after 45 visits, within the visits a history-based selector gives its most visited prompts and well beyond the 12 that uniform sampling gives each prompt in Math. The curves are closed-form; r is illustrative.

⟨1⟩5. $\langle \mathbb{E}S_n, u\rangle \geq \gamma\,\omega_{\min} n$ under the second assumption. PROOF: By ⟨1⟩1, $\langle \mathbb{E}S_n, u\rangle = \sum_k \mathbb{E}[\omega_k \langle \Pi_k C_k, u\rangle]$, and $\omega_k \geq \omega_{\min}$.

⟨1⟩6. Q.E.D. PROOF: By ⟨1⟩4 and ⟨1⟩5.

Under response-independent error the error term $-(\alpha+\beta)\nabla p$ is also linear in visits, but it lies along the true direction and only shrinks the true update. Measured in units of σ, the response-dependent drift is rn with $r = \gamma\omega_{\min}/\sigma$, while the root-mean-square size of the orthogonal component under response-independent error is at most $\sqrt{n}$; the drift exceeds twice that size once $n > (2/r)^2$. For $r = 0.3$ this means 45 visits. For $r = 0.1$ it means 401 (Figure 5). We do not estimate γ; the values of r are illustrative.

D.6 COROLLARY 1 AND CHANCE EVENTS

ASSUME: response-independent error ($a_i \equiv \alpha_i$ on wrong responses, $b_i \equiv \beta_i$ on correct ones) with $\alpha_i + \beta_i < 1$.

PROVE: (i) every parameter change moves q_i by $(1 - \alpha_i - \beta_i)$ times the change of p_i and the gap e_i by $-(\alpha_i + \beta_i)$ times it; (ii) the prompt's own step does not lower p_i to first order in expectation; (iii) a visit whose eight responses are all wrong yields $k^{\text{obs}} \geq 6$ with probability $\Pr(\text{Bin}(8, \alpha_i) \geq 6)$.

⟨1⟩1. α_i and β_i do not depend on θ. PROOF: $\alpha_i = \mathbb{E}[a_i \mid V = 0]$ with a_i constant on wrong responses, and likewise for β_i.

⟨1⟩2. $q_i(\theta) = \alpha_i + (1 - \alpha_i - \beta_i)p_i(\theta)$ and $e_i(\theta) = \alpha_i - (\alpha_i + \beta_i)p_i(\theta)$ for every θ. PROOF: Rearrange $q_i = p_i(1 - \beta_i) + (1 - p_i)\alpha_i$; the coefficients are constant by ⟨1⟩1.

⟨1⟩3. (i) holds. PROOF: Take differences in ⟨1⟩2. The identity itself holds for any error model at a fixed θ; only constant rates tie the changes together.

⟨1⟩4. (ii) holds. PROOF: By Proposition 3(ii), the expected first-order change of p_i is $\eta\,\omega_G(q_i)(1 - \alpha_i - \beta_i)\nabla p_i^\top H \nabla p_i \geq 0$.

⟨1⟩5. (iii) holds. PROOF: Given that all eight responses are wrong, each verdict is Bernoulli(α_i) and the verdicts are independent by (A1), so $k^{\text{obs}} \sim \text{Bin}(8, \alpha_i)$.

⟨1⟩6. Q.E.D. PROOF: A rise of e_i requires a fall of p_i by (i), which by (ii) comes from other prompts' updates, the noise of the prompt's own stochastic updates or second-order terms; a reward-hacking event, which combines $k^{\text{obs}} \geq 6$ with $k^{\text{true}} = 0$ after a visit with a correct response, otherwise arises only by chance, as in (iii).

The tail probability is

$$\Pr(\mathrm{Bin}(8,\alpha) \geq 6) = \sum_{k=6}^{8} \tbinom{8}{k} \alpha^k (1-\alpha)^{8-k},$$

which equals $1.23 \times 10^{-3},\ 2.09 \times 10^{-3}$ and 1.13×10^{-2} at $\alpha = 0.2,\ 0.22$ and 0.3.

The chance expectation. Corollary 1 gives the expected number of reward-hacking events that response-independent error produces by chance, which Section 3 subtracts from the measured count to obtain chance-adjusted events. An at-risk visit is a visit that follows a judge-informative visit with at least one correct response; an event is the first at-risk visit of a prompt with $k^{\mathrm{true}} = 0$ and $k^{\mathrm{obs}} \geq 6$. Write $t(\alpha) = \Pr(\mathrm{Bin}(8,\alpha) \geq 6)$.

ASSUME: response-independent error, and prompt i has at-risk visits $k = 1, \ldots, n_i$.
PROVE: the expected number of chance events on prompt i is at most $1 - \prod_{k \leq n_i}(1 - t(\alpha_i))$, and every all-wrong group of prompt i, at any earlier step, gives an unbiased estimate of α_i.

$\langle 1 \rangle 1$. α_i is the same at every step. PROOF: $\langle 1 \rangle 1$ of the proof of Corollary 1.

$\langle 1 \rangle 2$. At an at-risk visit with $k^{\mathrm{true}} = 0$ the event probability is $t(\alpha_i)$, and at one with $k^{\mathrm{true}} > 0$ it is zero. PROOF: $\langle 1 \rangle 5$ of the proof of Corollary 1, and the event definition.

$\langle 1 \rangle 3$. Given the responses, the verdicts at different visits are independent. PROOF: The judge randomness is drawn afresh for every verdict (A1).

$\langle 1 \rangle 4$. The probability of at least one event is at most $1 - \prod_{k \leq n_i}(1 - t(\alpha_i))$. PROOF: By $\langle 1 \rangle 2$ each visit has event probability at most $t(\alpha_i)$, independently by $\langle 1 \rangle 3$; only the first event counts.

$\langle 1 \rangle 5$. The share of accepted responses in an all-wrong group of prompt i has mean α_i. PROOF: Each verdict is Bernoulli(α_i) by $\langle 1 \rangle 1$ and (A1).

$\langle 1 \rangle 6$. Q.E.D. PROOF: By $\langle 1 \rangle 4$ and $\langle 1 \rangle 5$.

The chance expectation of a run sums the bound of $\langle 1 \rangle 4$ over its prompts, with α_i replaced by the prompt's false-positive rate over its own earlier all-wrong groups, shrunk toward a Beta$(0.59, 1.74)$ distribution. That distribution has mean 0.253, the pooled false-positive rate of the first 50 steps. It also reproduces 0.640, the probability that a group the verifier rejects in full is judge-informative. By $\langle 1 \rangle 2$, only all-wrong at-risk visits, those at which the verifier accepts none of the eight responses, can produce an event; of the at-risk visits defined in Section 2 (about 1,470 per uniform run and 4,100 to 4,300 per selector run; Appendix F), Table 14 lists these all-wrong ones, and the expectation sums over them. Under response-independent error the prompt's past rate estimates its future rate by $\langle 1 \rangle 1$, so measured events in excess of the expectation indicate false-positive rates that rose with training, which is the signature of response-dependent error (Proposition 3). Table 14 gives the expectation per run. Under uniform sampling measured events exceed it by 0.6 ± 0.1; for GRESO and MoPPS the chance-adjusted events are 10.8 ± 0.6 and 18.5 ± 0.7. Chance thus accounts for about two thirds of GRESO's events, through more visits to prompts the judge already over-credits, and the remaining third is the response-dependent part.

Two coarser references bracket this expectation. At a common false-positive rate of 0.22, chance explains only 0.1–0.5 events per run. With per-prompt rates drawn from the fitted Beta distribution and reweighted by $h(\alpha_i)$, the probability that an all-wrong group is judge-informative (a proxy for the at-risk condition), the per-visit event probability is $\mathbb{E}[h(\alpha)t(\alpha)]/\mathbb{E}[h(\alpha)] = 0.127$ (exact Beta moments), and counting every all-wrong at-risk visit gives the column "Beta, weighted" of Table 14. That column ignores both the first-occurrence rule and each prompt's own rate, which is why it exceeds the chance expectation for GRESO and MoPPS, whose all-wrong at-risk visits repeat on few prompts, and falls below it for uniform sampling, whose all-wrong at-risk visits fall on prompts with high rates.

Evidence of response dependence. A fixed wrong response is re-accepted on 0.68 ± 0.05 of re-checks, while the re-acceptance rate across different wrong responses to the same prompt is 0.49 ± 0.05 (Appendix C). With response-independent errors and per-prompt rates, both quantities equal $\mathbb{E}[\alpha_i^2]/\mathbb{E}[\alpha_i]$ over their prompts; the two measurements come from different samples (the re-check sample and the all-wrong training groups of steps 0–49), so the comparison is indicative rather than exact, but the same-response rate on the training serving stack, 0.727 ± 0.020, is also well

Table 14: Reward-hacking events and the number that response-independent error would produce by chance, per Math run under the Gemma-3-12B judge. All-wrong at-risk visits are at-risk visits (following a judge-informative visit with a correct response) at which the verifier accepts none of the eight responses; only these can produce an event. The chance expectation uses each prompt's own earlier false-positive rate and counts at most one event per prompt; the two references use a common rate of 0.22 or per-prompt rates from the fitted Beta distribution weighted by the probability that an all-wrong group is judge-informative, counting every all-wrong at-risk visit. Chance-adjusted events are measured events minus the chance expectation. GRESO and MoPPS ran twice in seeds 0 and 1 (runs a and b); the mean rows give mean ± SE over the three seeds, averaging the two runs within a seed first.

		all-wrong at-risk visits	measured events	common 0.22	Beta, weighted	chance expectation	chance-adjusted
Uniform	0	55	11	0.12	7.0	10.5	+0.5
	1	62	11	0.13	7.8	10.6	+0.4
	2	60	12	0.13	7.6	11.2	+0.8
	mean	59.0 ± 2.1	11.3 ± 0.3	0.13 ± 0.00	7.5 ± 0.3	10.8 ± 0.2	$+0.6 \pm 0.1$
GRESO	0a	190	31	0.40	24.1	20.3	+10.7
	0b	135	28	0.28	17.1	14.9	+13.1
	1a	172	23	0.36	21.8	17.6	+5.4
	1b	224	37	0.47	28.4	23.1	+13.9
	2	180	30	0.38	22.8	19.2	+10.8
	mean	180.2 ± 10.3	29.8 ± 0.2	0.38 ± 0.02	22.8 ± 1.3	19.1 ± 0.8	$+10.8 \pm 0.6$
MoPPS	0a	159	44	0.33	20.1	19.6	+24.4
	0b	96	22	0.20	12.2	12.1	+9.9
	1a	96	30	0.20	12.2	12.2	+17.8
	1b	97	32	0.20	12.3	12.0	+20.0
	2	110	33	0.23	13.9	13.6	+19.4
	mean	111.3 ± 9.0	32.3 ± 0.7	0.23 ± 0.02	14.1 ± 1.1	13.9 ± 1.1	$+18.5 \pm 0.7$

above 0.49. Figure 1(a) averages the prompt-runs with a reward-hacking event. On them the judge's mean passes rise from 3.15 to 6.42 of 8 across visits. The verifier's mean passes stay between 1.3 and 1.8. These prompt-runs are selected on a judged count of at least 6 at the event visit, so part of the judge's rise reflects that selection; the flat verifier rate, which the selection does not raise, is the part of this evidence that bears on Corollary 1. Interference from other prompts is bounded by the difficulty-matched replay, which trains GRESO's schedule on difficulty-matched prompts and records 11.0 ± 1.2 events, as many as uniform sampling (Table 1).

Relation to noise correction. Noise-corrected GRPO (El Mansouri et al., 2026) maps each verdict affinely with false-positive and false-negative rates that do not depend on the response, which divides the judged pass-rate gradient by $1 - \alpha - \beta$. It thereby restores the true term and scales the response-dependent term C_i by the same factor $1/(1 - \alpha - \beta)$, so it cannot remove it, and by Corollary 1 it cannot remove the events either. The affine map preserves unanimity, so GRESO's selection rule is unchanged, although the policy, and hence the realized visits, differ from GRESO's.

D.7 Observation 1

ASSUME: $c \in [0, 1]$ and a response y^* with $\nabla_\theta \pi_\theta(y^* \mid x_i) \neq 0$. Setting A has $V(y^*) = 0$ and $a_i(y^*) = c$; setting B has $V(y^*) = 1$ and $b_i(y^*) = 1 - c$; everything else is identical.

PROVE: the observations available to any procedure that uses the training judge (prompts, sampled responses and any number of verdicts on any of them, repeated queries included) have the same law in A and B; ∇p_i differs; no estimator T of ∇p_i built from the observations is unbiased in both; and $\max_{S \in \{A,B\}} \mathbb{E}_S \|T - \nabla p_i^S\|^2 \geq \frac{1}{4} \|\nabla_\theta \pi_\theta(y^* \mid x_i)\|^2$.

⟨1⟩1. $\Pr(J = 1 \mid x_i, y^*) = c$ in both settings. PROOF: In A the acceptance probability is $a_i(y^*) = c$; in B it is $1 - b_i(y^*) = c$.

⟨1⟩2. The observations have the same joint law in A and B. PROOF: They are generated by π_θ, the judge randomness and the acceptance probability $c(\cdot)$ (A1), which agree by ⟨1⟩1 and the assumption.
⟨1⟩3. $\nabla p_i^B - \nabla p_i^A = \nabla_\theta \pi_\theta(y^* \mid x_i) \neq 0$. PROOF: $p_i^B = p_i^A + \pi_\theta(y^* \mid x_i)$.
⟨1⟩4. T is biased in A or in B. PROOF: $\mathbb{E}_A T = \mathbb{E}_B T$ by ⟨1⟩2, and the targets differ by ⟨1⟩3.
⟨1⟩5. The minimax bound holds. PROOF: $\|\nabla p_i^A - \nabla p_i^B\| \leq \|T - \nabla p_i^A\| + \|T - \nabla p_i^B\|$, square with $(x+y)^2 \leq 2x^2 + 2y^2$, take expectations under the common law of ⟨1⟩2, and bound the sum of the two risks by twice their maximum.
⟨1⟩6. Q.E.D. PROOF: By ⟨1⟩2–⟨1⟩5.

The same argument applies to the expected exact-reward group gradient, which differs between the settings for the same reason, and for $c = 1$ setting B has $b_i(y^*) = 0$, so the construction remains valid if correct responses are never rejected.

Scope. Only the acceptance probability $c(y)$ of a response enters the expected update (Lemma 1); the judge's query-to-query randomness ζ averages out and does not accumulate. Re-asking the judge estimates $c(y)$ and nothing else. It can flag a wrong response only when its acceptance probability is lower than that of correct responses, which is an assumption about the false-negative rate rather than information in the verdicts; a wrong response that the judge accepts with high probability, and which is therefore re-accepted on most re-checks (a fixed wrong response is re-accepted on 0.68 ± 0.05 of them, Appendix C), is exactly the case the construction with c near one describes. Accordingly, a re-check by the same judge confirms most of the false positives it is meant to remove (Appendix G). The argument fails as soon as some observation has a different law in the two settings, as the verdict of a stronger judge, a reference answer or an exact verifier does; this is why the safeguards of Section 5 that detect errors rely on an outside reference. Observation 1 is an instance of the no-free-lunch argument of Everitt et al. (2017) for corrupted reward channels, specialized from the agent's policy in a Markov decision process to the policy gradient of a single judge, and a two-point argument in the sense of Le Cam (1973); Afrin & Shihab (2026) prove the analogue for the common-mode error of an evaluator ensemble, and identifiability of label noise fails without further assumptions for the same reason in classification (Scott et al., 2013).

D.8 PROOF OF THEOREM 1 AND THE VALUES OF THE BOUND

Setting. Fix a step t and condition on $\mathcal{F}_t$ throughout; we drop the index t. A selection design is a random subset $B \subseteq \{1, \ldots, N\}$ with inclusion probabilities $\rho_i = \Pr(i \in B) > 0$, joint inclusion probabilities $\rho_{ij} = \Pr(i \in B,\, j \in B)$ and expected size $n = \sum_i \rho_i$; each drawn prompt receives G rollouts. Given θ_t, the responses to x_i are sampled from $\pi_{\theta_t}(\cdot \mid x_i)$ independently of B and of the responses to other prompts. Let Φ_i be group i's summed per-token loss with its group-normalized advantages and Z the loss normalizer, fixed given $\mathcal{F}_t$, and let $g_i = (P/Z)\nabla_\theta \Phi_i$, so that uniform sampling's update is $P^{-1}\sum_{i \in B} g_i$, as in Proposition 2. An unbiased design uses

$$\hat{g} = \sum_{i \in B} \frac{g_i}{N\rho_i} = \frac{1}{P}\sum_{i \in B} w_i g_i, \qquad w_i = \frac{P}{N\rho_i}, \tag{3}$$

the Horvitz–Thompson estimator (Horvitz & Thompson, 1952) of $\bar{u} = N^{-1}\sum_i \bar{g}_i$, where $\bar{g}_i = \mathbb{E}[g_i]$ and $M_i = \mathbb{E}\|g_i\|^2$. Uniform sampling is the design with $\rho_i = P/N$, for which $w_i = 1$.

Lemma 2 (Second moment of a weighted design)**.** *For any design with $\rho_i > 0$,*

$$\mathbb{E}\|\hat{g}\|^2 = \underbrace{\frac{1}{N^2}\sum_i \frac{M_i}{\rho_i}}_{D(\rho)} + \frac{1}{N^2}\sum_{i \neq j} \frac{\rho_{ij}}{\rho_i \rho_j}\langle \bar{g}_i, \bar{g}_j \rangle. \tag{4}$$

Proof. ⟨1⟩1. $\hat{g} = \sum_i \mathbf{1}\{i \in B\}\, g_i/(N\rho_i)$. PROOF: Definition equation 3.
⟨1⟩2. $\|\hat{g}\|^2 = \sum_i \mathbf{1}\{i \in B\}\|g_i\|^2/(N\rho_i)^2 + \sum_{i \neq j} \mathbf{1}\{i \in B\}\mathbf{1}\{j \in B\}\langle g_i, g_j\rangle/(N^2\rho_i\rho_j)$. PROOF: Expand the square in ⟨1⟩1; $\mathbf{1}\{i \in B\}^2 = \mathbf{1}\{i \in B\}$.
⟨1⟩3. $\mathbb{E}[\mathbf{1}\{i \in B\}\|g_i\|^2] = \rho_i M_i$. PROOF: The rollouts of prompt i are independent of B.

⟨1⟩4. $\mathbb{E}[\mathbf{1}\{i \in B\}\mathbf{1}\{j \in B\}\langle g_i, g_j\rangle] = \rho_{ij}\langle \bar{g}_i, \bar{g}_j\rangle$ for $i \neq j$. PROOF: The rollouts of i and of j are independent of each other and of B.

⟨1⟩5. Q.E.D. PROOF: Take expectations in ⟨1⟩2 and substitute ⟨1⟩3 and ⟨1⟩4; $\rho_i M_i/(N\rho_i)^2 = M_i/(N^2\rho_i)$.

□

The first term, the design term $D(\rho)$, depends on the design through its marginal inclusion probabilities; the second depends on it through the joint inclusion probabilities and involves only the mean gradients. Under independent inclusions (Poisson sampling, $\rho_{ij} = \rho_i\rho_j$) the second term equals $\|\bar{u}\|^2 - N^{-2}\sum_i \|\bar{g}_i\|^2$, and under simple random sampling without replacement, $\rho_{ij} = P(P-1)/(N(N-1))$, the second moment becomes

$$\mathbb{E}\|\hat{g}_{\mathrm{U}}\|^2 = \frac{\bar{M}}{P} + \frac{N(P-1)}{P(N-1)}\Big(\|\bar{u}\|^2 - \frac{1}{N^2}\sum_i \|\bar{g}_i\|^2\Big), \qquad \bar{M} = \frac{1}{N}\sum_i M_i. \tag{5}$$

Relative to uniform sampling's design term $D_{\mathrm{U}} = \bar{M}/P$, the part $N(P-1)/(P(N-1)) \cdot N^{-2}\sum_i \|\bar{g}_i\|^2$ is at most $(P-1)/(N-1) < P/N$, since $\|\bar{g}_i\|^2 \leq M_i$ by Jensen's inequality (P/N is 4.0% in Math and 7.9% in Science), and the part involving $\|\bar{u}\|^2$ is of order $P\|\bar{u}\|^2/\bar{M}$, which is small when the per-group noise dominates the mean update. For a non-uniform fixed-size design the joint-inclusion term is not bounded by this argument, so Theorem 1 compares designs on $D(\rho)$ only.

Theorem 1 (formal statement)**.** *Let $n(\rho) = \sum_i \rho_i$. Every design with $\rho_i > 0$ satisfies*

$$n(\rho)\, D(\rho) \;\geq\; \frac{1}{N^2}\Big(\sum_i \sqrt{M_i}\Big)^2, \tag{6}$$

with equality if and only if $\rho_i \propto \sqrt{M_i}$ when every $M_i > 0$ (if some $M_i = 0$, the bound is approached as their $\rho_i \to 0$). Consequently, a design with $D(\rho) \leq D_{\mathrm{U}}$ has $n(\rho) \geq P\,v^\star$ with $v^\star = (N^{-1}\sum_i \sqrt{M_i})^2/(N^{-1}\sum_i M_i) \leq 1$, and it rolls out at least $v^\star$ times as many responses as uniform sampling. If the design may depend only on a sub-σ-algebra $\mathcal{G} \subseteq \mathcal{F}_t$, write $\tilde{M}_i = \mathbb{E}[M_i \mid \mathcal{G}]$. Then a design with $\mathbb{E}[D(\rho) \mid \mathcal{G}] \leq \mathbb{E}[D_{\mathrm{U}} \mid \mathcal{G}]$ has $n(\rho) \geq Pv^{\mathcal{G}}$ with $v^{\mathcal{G}} = (N^{-1}\sum_i \sqrt{\tilde{M}_i})^2/(N^{-1}\sum_i \tilde{M}_i)$, and

$$v^{\mathcal{G}} \;\geq\; \frac{\mathbb{E}\big[(N^{-1}\sum_i \sqrt{M_i})^2 \mid \mathcal{G}\big]}{\mathbb{E}\big[N^{-1}\sum_i M_i \mid \mathcal{G}\big]}. \tag{7}$$

Proof. ASSUME: $\rho_i > 0$ for every i, $n = \sum_i \rho_i$ and $D = N^{-2}\sum_i M_i/\rho_i$.

⟨1⟩1. $(\sum_i \sqrt{M_i})^2 \leq (\sum_i M_i/\rho_i)(\sum_i \rho_i)$. PROOF: Cauchy–Schwarz for the vectors $(\sqrt{M_i/\rho_i})_i$ and $(\sqrt{\rho_i})_i$, whose inner product is $\sum_i \sqrt{M_i}$.

⟨1⟩2. If every $M_i > 0$, equality in ⟨1⟩1 holds if and only if $\rho_i \propto \sqrt{M_i}$. PROOF: Equality holds if and only if $\sqrt{M_i/\rho_i} = c\sqrt{\rho_i}$ for one $c > 0$, that is, $\rho_i = \sqrt{M_i}/c$.

⟨1⟩3. $nD \geq N^{-2}(\sum_i \sqrt{M_i})^2$, which is equation 6. PROOF: Multiply ⟨1⟩1 by N^{-2}; the right side is $N^2 nD$.

⟨1⟩4. $D_{\mathrm{U}} = \bar{M}/P$. PROOF: Substitute $\rho_i = P/N$ into D.

⟨1⟩5. If $D \leq D_{\mathrm{U}}$, then $n \geq Pv^\star$. PROOF: By ⟨1⟩3, $n \geq N^{-2}(\sum_i \sqrt{M_i})^2/D \geq N^{-2}(\sum_i \sqrt{M_i})^2 P/\bar{M} = Pv^\star$, using ⟨1⟩4.

⟨1⟩6. $v^\star \leq 1$. PROOF: $(N^{-1}\sum_i \sqrt{M_i})^2 \leq N^{-1}\sum_i M_i$ by Cauchy–Schwarz.

⟨1⟩7. The design $\rho_i = Pv^\star\sqrt{M_i}/\sum_j \sqrt{M_j}$ has $n = Pv^\star$ and $D = D_{\mathrm{U}}$ whenever every $\rho_i \leq 1$. PROOF: Sum the ρ_i; equality holds in ⟨1⟩3 by ⟨1⟩2, so $D = N^{-2}(\sum_i \sqrt{M_i})^2/(Pv^\star) = D_{\mathrm{U}}$.

⟨1⟩8. If ρ is $\mathcal{G}$-measurable, $\mathbb{E}[D \mid \mathcal{G}] = N^{-2}\sum_i \tilde{M}_i/\rho_i$ and $\mathbb{E}[D_{\mathrm{U}} \mid \mathcal{G}] = N^{-1}\sum_i \tilde{M}_i/P$. PROOF: Linearity of conditional expectation; ρ_i is known given $\mathcal{G}$.

⟨1⟩9. If $\mathbb{E}[D \mid \mathcal{G}] \leq \mathbb{E}[D_{\mathrm{U}} \mid \mathcal{G}]$, then $n \geq Pv^{\mathcal{G}}$. PROOF: Steps ⟨1⟩1, ⟨1⟩3, ⟨1⟩4 and ⟨1⟩5 with $\tilde{M}_i$ in place of M_i, by ⟨1⟩8.

⟨1⟩10. $\mathbb{E}[(\sum_i \sqrt{M_i})^2 \mid \mathcal{G}] \le (\sum_i \sqrt{\tilde{M}_i})^2$.

⟨2⟩1. $\mathbb{E}[\sqrt{M_i M_j} \mid \mathcal{G}] \le \sqrt{\tilde{M}_i \tilde{M}_j}$ for every pair (i, j), including $i = j$. PROOF: Conditional Cauchy–Schwarz inequality applied to $\sqrt{M_i}$ and $\sqrt{M_j}$.

⟨2⟩2. Q.E.D. PROOF: Expand $(\sum_i \sqrt{M_i})^2 = \sum_{i,j} \sqrt{M_i M_j}$ and sum ⟨2⟩1 over all pairs.

⟨1⟩11. $\sum_i \tilde{M}_i = \mathbb{E}[\sum_i M_i \mid \mathcal{G}]$. PROOF: Linearity of conditional expectation.

⟨1⟩12. Q.E.D. PROOF: ⟨1⟩2, ⟨1⟩3, ⟨1⟩5, ⟨1⟩6 and ⟨1⟩9 give the first claims. For equation 7, divide ⟨1⟩10 by N^2 and both sides by $N^{-1} \sum_i \tilde{M}_i$, which equals the denominator of equation 7 by ⟨1⟩11.

□

The bound equation 6 is the Godambe–Joshi lower bound for Horvitz–Thompson estimation in survey sampling, attained with inclusion probabilities proportional to the standard deviation (Godambe & Joshi, 1965), and the square-root rule of importance-sampling SGD (Zhao & Zhang, 2015; Needell et al., 2014; Katharopoulos & Fleuret, 2018), including its minibatch form with capped probabilities (Csiba & Richtárik, 2018). VIP (Nguyen et al., 2026) and GVM-RAFT (Yao et al., 2025) use the same variance-minimizing principle to allocate rollouts across prompts in RLVR; our use differs in that the allocation is a keep decision over the pool, the weights keep the expected update unbiased, and the bound is evaluated for the information a history-based selector has. What is specific to GRPO is that zero-variance groups contribute no gradient.

Prompts that are never informative. If $M_i = 0$ for a fraction z_0 of the pool and $M_i = M$ for the rest, then $v^\star = ((1 - z_0)\sqrt{M})^2/((1 - z_0)M) = 1 - z_0$. A prompt has $M_i = 0$ only if it would contribute no gradient on any visit; a prompt that is unanimous on one visit but informative on a later one has $M_i > 0$, which is why the savings are smaller than the share of zero-variance groups on a single visit suggests.

The first-order model. We evaluate the bound under score homogeneity, which means that, given the rewards, the per-response score vectors have a common expected squared norm and are uncorrelated across responses; we call the result the first-order model. Then M_i is proportional to $\mathbb{E}[\sum_j \hat{A}_{ij}^2]$, and since a binary group has $\sum_j \hat{A}_{ij}^2 = G - 1$ when it is informative (up to the 10^{-4} in the denominator) and zero otherwise, $M_i \propto \mu_i$, the probability that prompt i's group is judge-informative. This is an assumption about the policy's score vectors that we do not test; it fixes only the relative sizes of the M_i. In Science the overlong penalty gives all-fail groups of unequal length a nonzero advantage, so some prompts with $\mu_i = 0$ still carry gradient; the true M_i are then less heterogeneous than μ_i, and the Science savings below are upper bounds.

Estimation. GRESO's information about a prompt is its streak state $s_{t,i}$, which records whether it has been visited and the length of its current run of consecutive all-pass or all-fail visits. We take $\mathcal{G}$ to be generated by the streak states and assume that prompts are exchangeable given their state, so that M_i depends on i only through $s_{t,i}$; the bound therefore covers designs whose propensity is a function of the streak state alone, not every design that could be built from GRESO's full record. For each 50-step window we estimate $\mu(s)$ as the fraction of visits in state s that were judge-informative in the uniform run of each seed, evaluate $v^{\mathcal{G}}$ on the distribution of states over the pool along the uniform run's trajectory, and report the mean $\pm$ SE over the three seeds (Table 16). The variance-calibrated keep rule is $\kappa(s) = \max\{\kappa_{\min}, \sqrt{\mu(s)/\mu_{\max}}\}$ with $\mu_{\max} = \max_s \mu(s)$ and $\kappa_{\min} = 0.3$, drawn and weighted as in Algorithm 1. This rule is a counterfactual evaluated on the uniform runs' records, and its value in Table 16 is its design term relative to uniform sampling's in the same first-order model. In Math the first 50 steps give only two visits per prompt, which carry almost no streak information. There the bound is 0.94–0.96, and the calibrated rule saves $4.6 \pm 0.2\%$. Table 15 gives the pooled values after step 50. The floor at 0.3 costs at most 3.2 points in Math and 0.1 points in Science, in every window and seed.

Table 15: Theorem 1 pooled over the steps after step 50, relative to uniform sampling (first-order model; the calibrated rule is a counterfactual evaluated on the uniform runs' records). Mean ± SE over three seeds.

domain	streak-state bound	calibrated rule	running IW-GRESO
Math	0.791 ± 0.001	0.794 ± 0.001	0.88 ± 0.002
Math, whole run	–	0.815 ± 0.001	0.89 ± 0.003
Science	0.948 ± 0.003	0.948 ± 0.002	1.09 ± 0.008

Table 16: Theorem 1 by 50-step window under the first-order model (score homogeneity), giving the rollouts needed for uniform sampling's design term, relative to uniform sampling (lower is cheaper). "Bound" is $v^{\mathcal{G}}$ for designs that see only the streak state; the calibrated rule is the square-root keep rule floored at 0.3, a counterfactual evaluated on the uniform runs' records; IW-GRESO is the running wrapper with GRESO's own propensity floored at 0.3. Mean ± SE over three seeds, first-order model fitted to the uniform runs.

domain	steps	bound (streak state)	calibrated rule	IW-GRESO
Math	50–99	0.828 ± 0.011	0.829 ± 0.011	0.904 ± 0.019
	100–149	0.788 ± 0.010	0.801 ± 0.007	0.920 ± 0.005
	150–199	0.763 ± 0.001	0.779 ± 0.005	0.870 ± 0.005
	200–249	0.739 ± 0.001	0.771 ± 0.001	0.862 ± 0.006
	250–299	0.752 ± 0.015	0.769 ± 0.019	0.870 ± 0.018
Science	0–49	0.981 ± 0.001	0.981 ± 0.001	1.040 ± 0.004
	50–99	0.955 ± 0.002	0.956 ± 0.002	1.112 ± 0.008
	100–149	0.948 ± 0.002	0.948 ± 0.002	1.120 ± 0.011
	150–199	0.933 ± 0.005	0.933 ± 0.005	1.101 ± 0.007

Keep probabilities. Table 17 compares the calibrated rule's keep probabilities after consecutive all-pass visits with GRESO's $\max(\phi_{\text{easy}}^{c}, 0.01)$. GRESO's values hold once its tuner has lowered ϕ_{easy} to its floor, which happens at step 68 in Math and at steps 56–62 in Science.

Table 17: Keep probability after c consecutive all-pass visits: the calibrated rule (mean ± SE over three seeds), GRESO once its tuner has converged, and their ratio.

domain	c	calibrated rule	GRESO	ratio
Math	1	0.64 ± 0.004	0.05	12.8 ± 0.1
	2	0.53 ± 0.011	0.01	53.3 ± 1.1
	3	0.34 ± 0.006	0.01	33.6 ± 0.6
Science	1	0.70 ± 0.011	0.05	14.0 ± 0.2
	2	0.58 ± 0.005	0.01	58.5 ± 0.5
	3	0.49 ± 0.011	0.01	48.8 ± 1.1

The running wrapper. IW-GRESO uses GRESO's keep propensity $(1-\text{easy skip})(1-\text{hard skip})$, floored at 0.3; once the tuner has converged this is 0.3 after any all-pass streak and one otherwise. Its design term per rollout is 0.89 ± 0.003 of uniform sampling's in Math and 1.09 ± 0.008 in Science (Table 15). The Math value is the favorable end of the range that alternative first-order models give (about ± 0.05). Because the wrapper draws P prompts at every step, it rolls out exactly as many responses as uniform sampling; what changes is their usefulness. In Math, its Kish effective sample size is 38.2 judge-informative groups per 1,000 rollouts, against uniform sampling's 32.8. That is 16% more. Counting only groups the verifier finds informative, which excludes spurious groups, it is 29.8 ± 0.3 against 27.9 ± 0.2 (7% more). In Science it is 63.8 against 67.0 judge-informative groups (5% fewer).

Fixed per-prompt propensities. A design that fixes each prompt's propensity in advance from its run-average informativeness, which no online selector knows, could save at most $38.7 \pm 0.6\%$

of the rollouts in Math and $8.6 \pm 0.1\%$ in Science by the same argument, evaluated in sample. The sampling noise of about 16 visits per prompt inflates the apparent heterogeneity; correcting for it, the Science value is about $6.5 \pm 0.1\%$. These are not bounds on adaptive designs; the streak-state bound above is the one that applies to a selector with GRESO's information. MoPPS's posterior carries more information than the streak state and is not covered by it.

Relation to DPPO. DPPO rescales a retained prompt by $|\mathcal{S}_t|/|\mathcal{Q}| \cdot 1/(1 - P_t(q))$, the inverse of its keep probability times a batch-size ratio (Zhu et al., 2026a), so its prompt-level rescaling and our weight $P/(N\rho_{t,i})$ coincide up to a constant factor. DPPO bounds the variance that its pruning adds (Zhu et al., 2026a, Theorem D.1). The factor is about 1.42 for pruning rates up to 0.7, and about 3.0 at a rate of 0.9. At the pruning rate used in our comparison its weights reach 10, whereas ours never exceed $1/\kappa_{\min} \approx 3.3$. Theorem 1 answers the complementary question. A design that rolls out $0.6P$ prompts per step on average has, by equation 6, a design term of at least $v^\star/0.6$ times uniform sampling's. With $v^\star \approx 0.79$ in Math, that is at least about 1.3 times. DPPO's speed-up therefore comes from cheaper steps at a higher per-step second moment, while the wrapper spends uniform sampling's rollouts at the second moments of Table 15; which is preferable depends on whether generation or optimizer steps dominate the cost.

D.9 Proof of Proposition 4 and its exactness conditions

Lemma 3 (Systematic sampling realizes the inclusion probabilities)**.** *Let $\rho_1, \dots, \rho_N \in (0, 1]$ with $\sum_i \rho_i = P$ an integer, let σ be any ordering of the pool, and let $c_k = \sum_{l \le k} \rho_{\sigma(l)}$ with $c_0 = 0$. Draw $u \sim \mathrm{Uniform}[0, 1)$ and select $\sigma(k)$ whenever some point $u + j$, $j \in \{0, \dots, P - 1\}$, lies in $[c_{k-1}, c_k)$. Then exactly P distinct prompts are selected and $\Pr(i \in B) = \rho_i$ for every i.*

Proof. ⟨1⟩1. The intervals $I_k = [c_{k-1}, c_k)$, $k = 1, \dots, N$, partition $[0, P)$. PROOF: $c_0 = 0$, $c_N = \sum_i \rho_i = P$, and c_k increases strictly because every $\rho_i > 0$.

⟨1⟩2. The P points $u + j$, $j = 0, \dots, P - 1$, lie in $[0, P)$, one in each interval $[j, j + 1)$. PROOF: $0 \le u < 1$.

⟨1⟩3. Each point lies in exactly one I_k. PROOF: ⟨1⟩1 and ⟨1⟩2.

⟨1⟩4. Each I_k contains at most one point. PROOF: I_k is half-open with length $\rho_{\sigma(k)} \le 1$, and the points are spaced by one.

⟨1⟩5. $|B| = P$ and no prompt is selected twice. PROOF: ⟨1⟩3 and ⟨1⟩4.

⟨1⟩6. I_k contains a point if and only if u lies in the image of I_k modulo one. PROOF: The points are $u + j$ with j an integer.

⟨1⟩7. That image has Lebesgue measure $\rho_{\sigma(k)}$. PROOF: The integer translates of an interval of length at most one do not overlap modulo one.

⟨1⟩8. Q.E.D. PROOF: ⟨1⟩5; and $\Pr(\sigma(k) \in B) = \rho_{\sigma(k)}$ by ⟨1⟩6, ⟨1⟩7 and $u \sim \mathrm{Uniform}[0, 1)$.

□

The ordering σ is a fresh random permutation at every step (Madow, 1949; Tillé, 2006); it changes the joint inclusion probabilities but not the marginal ones, which are all Proposition 4 uses. The inclusion probabilities themselves are $\rho_i = \min(1, \lambda\kappa_i)$, where λ solves $\sum_i \rho_i = P$ by water-filling, in which prompts with $\lambda\kappa_i > 1$ are fixed at one and λ is re-solved on the others until none exceeds one.

Proof of Proposition 4. ASSUME: $\kappa_{t,i}$ and Z_t are functions of $\mathcal{F}_t$; the draw uses randomness independent of the rollouts. PROVE: $\mathbb{E}[\nabla_\theta L \mid \mathcal{F}_t] = Z_t^{-1}(P/N) \sum_i \mathbb{E}[\nabla_\theta \Phi_i \mid \mathcal{F}_t]$ and $\Lambda = 1$.

⟨1⟩1. For $w > 0$, $\ell(\eta, wA) = w\,\ell(\eta, A)$, where $\ell(\eta, A) = -\min\{\eta A, \bar{\eta} A\}$ and $\bar{\eta} = \mathrm{clip}(\eta, 1 - \epsilon_{\mathrm{low}}, 1 + \epsilon_{\mathrm{high}})$. PROOF: Multiplying both arguments of a minimum by $w > 0$ preserves their order.

⟨1⟩2. The loss of group i with advantages $w_i \hat{A}_{ij}$ is $w_i \Phi_i$, where Φ_i uses the unweighted advantages. PROOF: ⟨1⟩1 token by token; the sampler-mismatch factor m_{ij} multiplies whole responses and commutes with w_i.

⟨1⟩3. $L = Z_t^{-1} \sum_{i=1}^N \mathbf{1}\{i \in B_t\}\, w_{t,i} \Phi_i$. PROOF: ⟨1⟩2 and the definition of the batch loss.

⟨1⟩4. Given $\mathcal{F}_t$, the quantities Z_t, $\rho_{t,i}$ and $w_{t,i}$ are constants, and $\mathbf{1}\{i \in B_t\}$ is independent of Φ_i. PROOF: The assumptions; the rollouts of x_i depend only on θ_t and x_i.

⟨1⟩5. $\Pr(i \in B_t \mid \mathcal{F}_t) = \rho_{t,i}$. PROOF: Lemma 3; water-filling gives $\sum_i \rho_{t,i} = P$ and $0 < \rho_{t,i} \leq 1$.

⟨1⟩6. $\mathbb{E}[\nabla_\theta L \mid \mathcal{F}_t] = Z_t^{-1} \sum_i \rho_{t,i} w_{t,i} \, \mathbb{E}[\nabla_\theta \Phi_i \mid \mathcal{F}_t]$. PROOF: Linearity, ⟨1⟩3, ⟨1⟩4 and ⟨1⟩5; the weights are treated as data, so ∇_θ does not act on them.

⟨1⟩7. $\rho_{t,i} w_{t,i} = P/N$. PROOF: $w_{t,i} = P/(N\rho_{t,i})$.

⟨1⟩8. The right side of ⟨1⟩6 is the expected gradient of the design with $\rho_{t,i} = P/N$, $w_{t,i} = 1$ and the same Z_t. PROOF: Apply ⟨1⟩6 to that design and compare with ⟨1⟩7.

⟨1⟩9. $X_{t,i} = P/N$ for all i, t, so $\Lambda = \max_i \sum_{t \leq T} X_{t,i}/(TP/N) = 1$. PROOF: $X_{t,i} = \rho_{t,i} w_{t,i}$ and ⟨1⟩7.

⟨1⟩10. Q.E.D. PROOF: ⟨1⟩6, ⟨1⟩7, ⟨1⟩8 and ⟨1⟩9.

□

Algorithm 1 Importance-weighted selection, one training step t. The selector is used only as a state tracker; its own keep or top-P draw is never executed.

Require: pool $\{x_i\}_{i=1}^N$; prompts per step P; group size G; floor $\kappa_{\min} = 0.3$; clip range $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.2, 0.28)$; policy π_θ; normalizer average $\bar{Z}$ (undefined before the first step); the selector's state, which gives each prompt a keep propensity $\kappa_{t,i}$. For GRESO, $\kappa_{t,i} = \max(\phi_{\text{easy}}^{c^+_{t,i}}, 0.01) \max(\phi_{\text{hard}}^{c^-_{t,i}}, 0.01)$, where $c^\pm_{t,i}$ is the length of the prompt's current run of all-pass (all-fail) visits, at most one of them nonzero, and $\phi_{\text{easy}}, \phi_{\text{hard}}$ are GRESO's tuned base keep rates. For IW-MoPPS, $\kappa_{t,i} = \Pr(0.3 \leq q \leq 0.7)$ for $q \sim \text{Beta}(1 + k^+_{t,i}, 1 + k^-_{t,i})$, with $k^\pm_{t,i}$ the prompt's accepted and rejected response counts so far.

if $t > 1$ **then**
  $Z_t \leftarrow \bar{Z}$ ▷ normalizer fixed before the draw
end if
for $i = 1, \ldots, N$ **do**
  $\kappa_i \leftarrow \min\{1, \max\{\kappa_{\min}, \kappa_{t,i}\}\}$
end for
$C \leftarrow \emptyset$ ▷ water-filling for $\rho_i = \min(1, \lambda\kappa_i)$ with $\sum_i \rho_i = P$
repeat
  $\lambda \leftarrow (P - |C|)/\sum_{i \notin C} \kappa_i$; $C \leftarrow C \cup \{i \notin C : \lambda\kappa_i > 1\}$
until no prompt was added to C
$\rho_i \leftarrow 1$ for $i \in C$ and $\rho_i \leftarrow \lambda\kappa_i$ otherwise
draw a uniform random permutation σ of $\{1, \ldots, N\}$ and $u \sim \text{Uniform}[0, 1)$
$B_t \leftarrow \{\sigma(k) : \exists j \in \{0, \ldots, P-1\}, \ \sum_{l<k} \rho_{\sigma(l)} \leq u + j < \sum_{l \leq k} \rho_{\sigma(l)}\}$ ▷ $|B_t| = P$
for $i \in B_t$ **do**
  sample $y_{i1}, \ldots, y_{iG} \sim \pi_\theta(\cdot \mid x_i)$ and obtain the judge's verdicts $r_{i1}, \ldots, r_{iG}$
  $\bar{r}_i \leftarrow G^{-1} \sum_j r_{ij}$; $s_i \leftarrow$ sample standard deviation of $r_{i1}, \ldots, r_{iG}$
  $\hat{A}_{ij} \leftarrow w_i(r_{ij} - \bar{r}_i)/(s_i + 10^{-4})$ with $w_i = P/(N\rho_i)$, for $j = 1, \ldots, G$
  $T_{t,i} \leftarrow$ number of completion tokens in $y_{i1}, \ldots, y_{iG}$
end for
if $t = 1$ **then**
  $Z_1 \leftarrow \sum_{i \in B_1} w_i T_{1,i}$; $\bar{Z} \leftarrow Z_1$ ▷ no history at the first step
end if
$L \leftarrow Z_t^{-1} \sum_{i \in B_t} \sum_{j=1}^G m_{ij} \sum_\ell - \min\{\eta_{ij\ell}\hat{A}_{ij}, \ \bar{\eta}_{ij\ell}\hat{A}_{ij}\}$ with $\bar{\eta}_{ij\ell} = \text{clip}(\eta_{ij\ell}, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})$, where $\eta_{ij\ell}$ is the ratio of the current to the sampling policy's probability of token ℓ of y_{ij} and $m_{ij} = \nu_{ij}\mathbf{1}\{\nu_{ij} \leq 3\}$ is the sampler-mismatch correction (Appendix B)
take one optimizer step on L
if $t > 1$ **then**
  $\bar{Z} \leftarrow 0.8\,\bar{Z} + 0.2 \sum_{i \in B_t} w_i T_{t,i}$
end if
update the selector's state with the unweighted verdicts of B_t (GRESO's streak counters and tuner, or MoPPS's posterior); the weights enter only the loss

Exactness conditions. The identity holds under the following conditions, which the implementation enforces. First, $\kappa_{t,i}$ must be a function of $\mathcal{F}_t$, which holds because the selector's state includes every visit up to step $t-1$, including updates that arrive asynchronously, before the pool is scored. Second, the normalizer is fixed before the draw. Z_t is an exponential moving average, with rate 0.2, of the previous steps' weighted token counts $\sum_{i \in B_s} w_{s,i} T_{s,i}$, where $T_{s,i}$ is the number of completion tokens of group i; this weighted count is unbiased for uniform sampling's expected token count. At the first step there is no history, and the realized weighted count is used; this is the only step at which the identity is not exact. A normalizer equal to the realized unweighted token count would divide an unbiased numerator by a random denominator, which is biased. Third, the weights multiply the advantages after group normalization; weighting the rewards would change the normalization instead of the group's share of the update. Fourth, the trained set must be the drawn set. Post-rollout filtering, visit caps, cooldowns and retirements set some inclusion probabilities to zero or select among drawn groups, so the implementation refuses these combinations. Fifth, the identity holds for the clipped loss at any value of the ratio η; in our runs each batch is used for a single optimizer step, so $\eta = 1$ where the gradient is taken and the clip does not bind. Sixth, floating-point rounding can shift the last interval of the systematic draw; the implementation then completes the draw from the permutation order and logs the event. The proposition concerns the expected gradient at each step. Adam's update is a nonlinear function of the gradient history, so the expected parameter step and the trajectory differ from uniform sampling's at second order, and the wrapper changes the variance of the update (Theorem 1). The comparison design includes each prompt with probability P/N at every step; our uniform arm instead permutes the pool each epoch, which gives the same expected visits but different conditional inclusion probabilities within an epoch, and it normalizes by realized tokens. MoPPS draws the top P prompts by Thompson-sampled pass rate, which has no closed-form inclusion probability, so IW-MoPPS replaces that draw with Algorithm 1 and uses as $\kappa_{t,i}$ the posterior mass of the prompt's pass rate inside MoPPS's target band $[0.3, 0.7]$ (0.4 for a prompt never visited). After flooring, these propensities are nearly uniform, so IW-MoPPS serves as a control of the implementation rather than as a second selector.

D.10 The visit bound and triggered verification

The procedure. Let $n_{t,i}$ be the number of times prompt i has been trained on before step t. The visit bound allows a prompt at most $C_t = \Lambda_{\max} \max(1, tP/N)$ visits, with $\Lambda_{\max} = 1.5$, once it has at least eight visits; the selector's own rule decides everything below the bound. With the bound alone, a prompt over the bound is skipped, and if fewer than P prompts remain, the least visited of the skipped candidates and then uniformly drawn pool prompts fill the batch. Triggered verification changes what happens at the bound (Algorithm 2). A prompt over the bound is kept and marked for verification. On its next visit on which the judge accepts at least one response, an independent verifier (the exact verifier in Math) checks the accepted responses. If the verifier accepts none of them, the visit counts as exploited, and two consecutive exploited visits retire the prompt for the rest of training. If the verifier accepts at least one, the prompt is exempt from the bound until its visit count doubles, after which it is verified again. The training loss is unchanged and unweighted.

Algorithm 2 The visit bound with triggered verification, one training step t (GRESO as the selector).

Require: selector with its own keep rule; bound $\Lambda_{\max} = 1.5$; minimum of eight visits; verifier V; visit counts n_i; retired set R; pending set Q; exemption thresholds e_i (initially 0); exploited-visit counts a_i (initially 0)

$C_t \leftarrow \Lambda_{\max} \max(1, tP/N)$
$S \leftarrow$ the candidates the selector proposes for step t, without the prompts in R
for $i \in S$ with $n_i \geq e_i$, $n_i \geq 8$ and $n_i > C_t$ **do**
    add i to Q ▷ with the bound alone: remove i from S instead
end for
if $|S| < P$ **then**
    add the least visited skipped candidates, then uniformly drawn prompts not in R, until $|S| = P$
end if
$B_t \leftarrow$ the first P prompts of S in the selector's order
roll out G responses per prompt in B_t, score them with the judge and take the usual unweighted GRPO step
for $i \in B_t$ **do**
    $n_i \leftarrow n_i + 1$
    if $i \in Q$ and the judge accepted at least one response **then**
        $k \leftarrow$ number of those responses that V accepts ▷ one verifier check per accepted response
        if $k = 0$ **then**
            $a_i \leftarrow a_i + 1$; **if** $a_i = 2$ **then** move i from Q to R
        else
            $a_i \leftarrow 0$; remove i from Q; $e_i \leftarrow 2n_i$
        end if
    end if
end for
update the selector's state with the verdicts of B_t

Why the combination works. The bound caps every prompt's cumulative exposure near $\Lambda_{\max}$ times uniform sampling's, so by Proposition 2 the first-order bias from any set of prompts the judge errs on is at most about $\Lambda_{\max}$ times uniform sampling's worst case; the bound alone lowers GRESO's reward-hacking events from 29.8 to 15.0 in Math. It does so without judge information, which is also its limitation, since it caps prompts the policy is genuinely learning from as well. Verification uses the bound only to decide where to look. Because response-dependent error grows with visits (Proposition 3), the prompts that reach the bound are enriched for the judge's errors. Retiring those that the verifier rejects bounds their budget at the bound, and exempting those it confirms leaves the selector free to concentrate on genuine prompts. With verification the judged-minus-true gap falls to 0.012 ± 0.001, below uniform sampling's 0.040 ± 0.002. Events fall to 17.3, against 11.3 for uniform sampling. The cost is 416 verified visits over 9,600 trained groups, 4.3% of the groups, each verified visit checking only the responses the judge accepted. Because a retired or skipped prompt has inclusion probability zero, neither change can be combined with the importance-weighted wrapper, whose floor on κ is the compatible form of a bound.

Remark 1 (Caps bound worst-case harm; uniform allocation minimizes it)**.** Let $x_i \geq 0$ be prompt i's cumulative weighted exposure, with $\sum_i x_i = N\bar{x}$ fixed, and suppose the harm from a prompt on which the judge errs is $h(x_i)$ with h increasing and convex. For an unknown set $\mathcal{E}$ of m such prompts, a cap $x_i \leq c\bar{x}$ bounds the worst-case harm $\max_{|\mathcal{E}|=m} \sum_{i\in\mathcal{E}} h(x_i)$ by $m\,h(c\bar{x})$, and uniform allocation, $x_i = \bar{x}$, minimizes the worst-case harm, which is then $m\,h(\bar{x})$.

Proof. ASSUME: h is increasing and convex, $\sum_i x_i = N\bar{x}$ and $|\mathcal{E}| = m$.

⟨1⟩1. Under the cap, $\sum_{i\in\mathcal{E}} h(x_i) \leq m\,h(c\bar{x})$ for every $\mathcal{E}$. PROOF: Each $x_i \leq c\bar{x}$ and h is increasing.

⟨1⟩2. For any allocation, $\max_{|\mathcal{E}|=m} \sum_{i\in\mathcal{E}} h(x_i) \geq (m/N) \sum_i h(x_i)$. PROOF: The m largest of N numbers average at least their mean.

⟨1⟩3. $N^{-1} \sum_i h(x_i) \geq h(\bar{x})$. PROOF: Jensen's inequality for the convex h.

⟨1⟩4. The uniform allocation $x_i = \bar{x}$ has worst-case harm $m\,h(\bar{x})$. PROOF: Every set of m prompts gives $m\,h(\bar{x})$.

⟨1⟩5. Q.E.D. PROOF: ⟨1⟩1 is the bound; ⟨1⟩2–⟨1⟩4 show that no allocation has worst-case harm below $m\,h(\bar{x})$.

□

A cap is therefore a backstop against concentration rather than an optimum. Only step ⟨1⟩3 uses convexity; the bound $m\,h(c\bar{x})$ needs only monotonicity.

D.11 THE RANDOM AUDIT, ITS ESTIMATOR AND ITS COST

Suppose a reference judge is available at a higher cost. After a group's responses and training-judge verdicts $J_i = (J_{i1}, \ldots, J_{iG})$ are fixed, an independent coin $I_i \sim \mathrm{Bernoulli}(p_i)$ decides whether the group is also graded by the reference, giving verdicts J_i^{ref}. Because the clipped per-token loss is not linear in the advantage, we correct its positive and negative parts separately. Write $\bar{\eta} = \mathrm{clip}(\eta, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})$. For all real η and A, $\min(\eta A, \bar{\eta} A) = A^+ \min(\eta, \bar{\eta}) - A^- \max(\eta, \bar{\eta})$ with $A^\pm = \max(\pm A, 0)$, so a group's loss is linear in the vector (A^+, A^-) of its split advantages. The audited split advantages are

$$\tilde{A}_{ij}^{\pm} = A_{ij}^{\pm}(J_i) + \frac{I_i}{p_i}\big(A_{ij}^{\pm}(J_i^{\mathrm{ref}}) - A_{ij}^{\pm}(J_i)\big), \tag{8}$$

where $A_{ij}(\cdot)$ is the group-normalized advantage computed from the given verdicts, and they enter the two-branch loss $-\sum_j[\tilde{A}_{ij}^{+} m_{ij}^{+} - \tilde{A}_{ij}^{-} m_{ij}^{-}]$ with $m_{ij}^{+} = \sum_\ell \min(\eta_{ij,\ell}, \bar{\eta}_{ij,\ell})$ and $m_{ij}^{-} = \sum_\ell \max(\eta_{ij,\ell}, \bar{\eta}_{ij,\ell})$ over the response's tokens. Substituting the corrected scalar advantage into the usual clipped loss would not be unbiased, because the minimum does not commute with the expectation over the coin when the corrected advantage can change sign.

Lemma 4 (Random audit)**.** *If the coins are drawn after the responses, verdicts and audit probabilities are fixed, independently of the reference verdicts, with $p_i > 0$, and the normalizer does not depend on the verdicts, then the expected loss and gradient equal those of training on the reference's verdicts for the same batch, and the excess second moment of the gradient is $\sum_i (1-p_i) p_i^{-1} \|\nabla L_i^{\mathrm{ref}} - \nabla L_i\|^2 / Z^2$, where L_i^{ref} and L_i are group i's losses under the reference's and the training judge's verdicts. The identity holds conditionally at every step, so a selector may use earlier audits to choose later prompts.*

Proof. ⟨1⟩1. $\min(\eta A, \bar{\eta} A) = A^+ \min(\eta, \bar{\eta}) - A^- \max(\eta, \bar{\eta})$ for all real $\eta, \bar{\eta}, A$. PROOF: For $A \geq 0$ both sides equal $A \min(\eta, \bar{\eta})$; for $A < 0$ both equal $A \max(\eta, \bar{\eta})$.

⟨1⟩2. A group's loss is linear in its split advantages (A^+, A^-), with coefficients $m_{ij}^{\pm}$ that do not depend on the verdicts. PROOF: Sum ⟨1⟩1 over tokens; η and $\bar{\eta}$ depend only on the policy.

⟨1⟩3. $\mathbb{E}[\tilde{A}_{ij}^{\pm} \mid \text{responses}, J_i, J_i^{\mathrm{ref}}] = A_{ij}^{\pm}(J_i^{\mathrm{ref}})$. PROOF: In equation 8 only I_i is random, I_i is independent of the verdicts, and $\mathbb{E}[I_i/p_i] = 1$.

⟨1⟩4. The expected loss and gradient equal those under the reference's verdicts. PROOF: ⟨1⟩2, ⟨1⟩3, linearity, and a normalizer that does not depend on the verdicts; the coin weights are treated as data.

⟨1⟩5. The audited gradient minus the reference gradient is $Z^{-1} \sum_i (I_i/p_i - 1)(\nabla L_i^{\mathrm{ref}} - \nabla L_i)$. PROOF: Substitute equation 8 into the linear loss of ⟨1⟩2.

⟨1⟩6. Its second moment is $\sum_i (1-p_i) p_i^{-1} \|\nabla L_i^{\mathrm{ref}} - \nabla L_i\|^2 / Z^2$. PROOF: The terms of ⟨1⟩5 are independent with mean zero, and $\mathrm{Var}(I_i/p_i) = (1-p_i)/p_i$.

⟨1⟩7. Q.E.D. PROOF: ⟨1⟩4 and ⟨1⟩6; conditioning on the history at each step, with fresh coins, gives the statement at every step.

□

Under score homogeneity with a common rate p, the audited gradient's second moment about the reference gradient is $\mathcal{V}(1+(1-p)d/p)$, with $\mathcal{V} = \mathbb{E}\|A(J^{\mathrm{ref}})\|^2$ and $d = \mathbb{E}\|A(J^{\mathrm{ref}}) - A(J)\|^2 / \mathcal{V}$ per

group, so a noise multiplier k requires $p = d/(d + k - 1)$. We compute d exactly for $G = 8$ by enumerating all pairs of reference and judge verdict patterns, with independent reference verdicts at pass rate p^{ref} and the judge's false-positive and false-negative rates measured against the reference. The rubric judge has false-positive rate 0.213 and false-negative rate 0.163 on the calibration set, and the on-policy reference pass rate is 0.76–0.81. A multiplier of $k = 1.5$ then requires $p = 0.65 \pm 0.007$, so about two thirds of all groups must be graded by the reference; over reference pass rates from 0.3 to 0.9 the rate ranges from 0.60 to 0.74. The Math judge has on-policy false-positive rate 0.22–0.25 and false-negative rate 0.02, with verifier pass rate 0.78–0.80. It requires $p = 0.45 \pm 0.006$, and 0.38–0.52 over the plausible ranges of these rates; heterogeneity of the rates across prompts raises it. In Math the auditor is the exact verifier, which costs almost nothing, so auditing every group, that is, training on the verifier, is the cost-optimal choice. More generally, with generation cost c_{gen} and reference cost c_{ref} per response in units of one training-judge call, the product of cost and second moment $(c_{\mathrm{gen}} + 1 + p\, c_{\mathrm{ref}})(1 + (1 - p)d/p)$ is minimized at $p^\star = \sqrt{(c_{\mathrm{gen}} + 1)d/(c_{\mathrm{ref}}(1 - d))}$ (capped at one), and auditing beats grading every group with the reference only if $d < 1$ and the reference is much more expensive than the training judge; with d close to one, as for the rubric judge, the saving in reference calls is small. Unlike the visit-triggered audit of Section 5, which spends verification where the selector concentrates and changes which prompts are trained, the random audit changes the target of the update and leaves the selection untouched.

D.12 The answer-swap reward

Consider items whose answer lies in a known finite set $\mathcal{O}_x$ with exactly one correct element $o^\star$ (the options of a multiple-choice item; for Math, a candidate set containing the correct value). Write a response as (z, o), its reasoning z and final answer o, and let $J(z, o) \in \{0, 1\}$ be the judge's verdict. Draw o' uniformly from $\mathcal{O}_x \setminus \{o\}$ and use the reward

$$\hat{r}(z, o) = J(z, o) - J(z, o'). \tag{9}$$

Lemma 5 (Answer swap)**.** *(i)* $\mathbb{E}[\hat{r} \mid z, o] = J(z, o) - (|\mathcal{O}_x| - 1)^{-1} \sum_{o' \neq o} J(z, o')$. *(ii) If* $\Pr(J(z, o) = 1) = \psi(z) + \zeta(z)\mathbf{1}\{o = o^\star\}$*, a leniency* ψ *that does not depend on the answer plus a sensitivity* $\zeta \geq 0$*, then* $\mathbb{E}\hat{r} = \zeta(z)\big(\mathbf{1}\{o = o^\star\} - \mathbf{1}\{o \neq o^\star\}/(|\mathcal{O}_x| - 1)\big)$*, so the leniency cancels for every response. (iii) For a deterministic judge,* $\mathbb{E}[\hat{r} \mid z, o] = 1$*, the maximum, holds if and only if* $J(z, o) = 1$ *and* $J(z, o') = 0$ *for every* $o' \neq o$*; a response with a wrong answer attains it only if the judge accepts that answer and rejects every alternative, including the correct one, given the same reasoning.*

Proof. ⟨1⟩1. (i) holds. PROOF: o' is uniform on $\mathcal{O}_x \setminus \{o\}$, so $\mathbb{E}[J(z, o') \mid z, o]$ is the average of $J(z, o')$ over $o' \neq o$.

⟨1⟩2. Under the model of (ii), the average of $\Pr(J(z, o') = 1)$ over $o' \neq o$ is $\psi(z) + \zeta(z)\mathbf{1}\{o \neq o^\star\}/(|\mathcal{O}_x| - 1)$. PROOF: If $o \neq o^\star$, exactly one of the $|\mathcal{O}_x| - 1$ alternatives is $o^\star$; if $o = o^\star$, none is.

⟨1⟩3. (ii) holds. PROOF: Subtract ⟨1⟩2 from $\Pr(J(z, o) = 1) = \psi(z) + \zeta(z)\mathbf{1}\{o = o^\star\}$; $\psi(z)$ cancels.

⟨1⟩4. (iii) holds. PROOF: With binary deterministic verdicts, (i) equals one if and only if $J(z, o) = 1$ and the average of the $J(z, o')$ is zero, that is, every $J(z, o') = 0$.

⟨1⟩5. Q.E.D. PROOF: ⟨1⟩1, ⟨1⟩3 and ⟨1⟩4.

□

A cue that raises the judge's verdict for every answer is therefore subtracted out. The reward is not free of exploitation, however. By (i), a wrong answer earns a positive expected reward whenever the judge accepts it more often than it accepts the average alternative, and the maximal reward whenever the judge accepts it and rejects every alternative, including the correct one, an error that no query to the same judge can reveal (Observation 1). More importantly, a judge that checks whether the reasoning supports the final answer, as reasoning judges typically do, rejects almost every swapped answer, so $J(z, o') \approx 0$ and $\hat{r} \approx J(z, o)$; model (ii) excludes this case, and in it the swap removes nothing. The reward also requires a deterministic judge for (iii), a finite answer set containing the correct answer (for Math this set must be built from the reference answer), and one extra judge call per response; it applies to every Math prompt but to only 10.5% of the rubric items (20 of 190 with

an extractable unique option). We analyze the reward but do not train with it, since whether the false positives the policy exploits in Math survive an answer swap depends on the judge. Probes that delete parts of a response do not have property (iii), since a policy can spread a judge cue over the whole response so that no deletion removes it.

D.13 Floor sweep

The floor keeps every inclusion probability at least $\kappa_{\min} P/N$ and every weight at most $1/\kappa_{\min}$ (3.3 at $\kappa_{\min} = 0.3$). It protects against a misestimated propensity rather than being part of the optimum. Weighting GRESO's own draws by their inverse propensities with GRESO's floor of 0.01 would leave 3.6–4.9 effective groups per step late in Math training (Kish effective sample size, $(\sum w)^2 / \sum w^2$ over the judge-informative groups; Kish, 1965), against 6–8 judge-informative groups per step under uniform sampling. Table 18 varies the floor.

Table 18: Floor sweep. Second moment per rollout relative to uniform sampling (first-order model fitted to the uniform runs), realized Kish effective sample size over judge-informative groups per 1,000 rollouts (32.8 in Math and 67.0 in Science under uniform sampling), and Kish effective sample size over all drawn groups divided by P. The calibrated rule's rows are a counterfactual evaluated on the uniform runs' records. Mean $\pm$ SE over three seeds.

domain	design	floor	second moment	Kish ESS / 1,000	Kish ESS / P
Math	IW-GRESO	0.1	1.41 ± 0.002	39.6 ± 0.3^a	0.49 ± 0.000
		0.3	0.89 ± 0.003	38.2 ± 0.3	0.79 ± 0.001
		0.5	0.86 ± 0.001	37.6 ± 0.2	0.92 ± 0.000
	IW-MoPPS	0.1	1.12 ± 0.008	32.3 ± 0.1	0.66 ± 0.000
		0.3	0.92 ± 0.004	35.5 ± 0.2	0.92 ± 0.001
		0.5	0.95 ± 0.002	35.1 ± 0.1	0.98 ± 0.000
	calibrated rule	0.1	0.81 ± 0.000	40.5 ± 0.4	0.83 ± 0.001
		0.3	0.82 ± 0.001	40.3 ± 0.5	0.86 ± 0.003
		0.5	0.85 ± 0.001	37.9 ± 0.3	0.93 ± 0.001
Science	IW-GRESO	0.1	1.84 ± 0.016	57.3 ± 0.3	0.49 ± 0.001
		0.3	1.09 ± 0.008	63.8 ± 0.6	0.76 ± 0.002
		0.5	0.98 ± 0.003	68.1 ± 0.5	0.91 ± 0.000
	IW-MoPPS	0.1	1.57 ± 0.009	48.6 ± 0.2	0.52 ± 0.003
		0.3	1.05 ± 0.003	63.5 ± 0.4	0.84 ± 0.003
		0.5	0.99 ± 0.001	67.8 ± 0.5	0.95 ± 0.001
	calibrated rule	0.1	0.95 ± 0.003	69.6 ± 0.4	0.94 ± 0.002
		0.3	0.95 ± 0.002	69.8 ± 0.5	0.94 ± 0.002
		0.5	0.95 ± 0.002	69.6 ± 0.6	0.96 ± 0.001

[a] At floor 0.1 a few heavy weights dominate each step and bias the per-step Kish estimator upward; the first-order value is 23.3 ± 0.2.

A low floor lets GRESO's near-zero propensities through and raises the second moment above uniform sampling's in both domains, since the few drawn prompts with small propensity receive large weights. A floor of 0.5 on GRESO's own propensity gives the lowest second moment of the running configurations (0.86 in Math and 0.98 in Science), while the calibrated rule is nearly insensitive to its floor, because its propensities rarely reach it. Every other run in the paper uses 0.3, the value fixed in the analysis plan before training. The floor also limits concentration by itself, without the weights; Section 6 reports that control, the same floored draw trained without reweighting. In every window, the realized Kish values at floor 0.3 lie 2–5% above the first-order values in Science and 6–9% above in Math (Figure 3b for Math, Appendix F for Science), which is the evidence that the model behind Table 16 describes the runs.

E Experimental details

E.1 Training and evaluation setup

Policies and training. Every policy is trained from a base checkpoint, Qwen3-1.7B-Base in both domains and Qwen3-8B-Base in the Science block (Yang et al., 2025), with full fine-tuning in TRL

(von Werra et al., 2020) and a vLLM sampler co-located with the trainer on the same node. Each step draws $P = 32$ prompts and $G = 8$ responses per prompt, sampled at temperature 1.0. It normalizes the rewards within each group and takes one optimizer step on the token-level clipped loss without a KL term; Appendix B.2 writes the loss exactly as trained. Table 19 lists the hyperparameters, which are the same for every method within a domain. A Math run lasts 300 steps, so that uniform sampling visits each prompt of the 800-prompt pool 12 times. A Science run lasts 200 steps, or about 16 visits per prompt. Uniform sampling draws prompts without replacement within an epoch; since the Science pool size is not a multiple of P, its visit amplification there is 1.08 rather than 1.

Math. The pool comes from the training split of MATH (Hendrycks et al., 2021), with the MATH-500 problems (Lightman et al., 2024) excluded. We admitted a problem when Qwen3-1.7B-Base solved some but not all of 8 samples at a 1,024-token budget. We kept 800 admitted problems in round-robin order over the five difficulty levels. The hard set $\mathcal{H}$ holds the 359 pool prompts the base policy solves at most twice in eight samples. Training prompts are the bare problem text in Qwen3's chat template, without an answer-format instruction. The training judge, Gemma-3-12B-it (Gemma Team et al., 2025), sees the problem and the response but no reference answer; it reasons briefly and ends with a verdict of correct or incorrect, sampled at temperature 0.6 with at most 2,400 tokens (Appendix C gives the prompt and its error rates). The exact verifier normalizes the final answer symbolically, falls back to string matching, and compares it with the reference; it grades every training response for the analysis, provides the reward only in the exact-verifier runs, and grades every evaluation. MATH held-out consists of 500 further admitted problems, 100 per level and disjoint from the pool, decoded greedily with 1,024 tokens. The held-out suite (AMC 2023, Minerva Math and OlympiadBench; Lewkowycz et al., 2022; He et al., 2024) is decoded greedily with 2,048 tokens and graded by the same verifier.

Science. The pool comes from the training split of RaR-Science (Gunjal et al., 2026). We kept questions and reference answers of at most 512 tokens with 7 to 12 rubric criteria. We removed near-duplicates of evaluation items (token Jaccard similarity of at least 0.5) and sampled eight base-policy responses per question. We then admitted the 406 questions on which the reference judge accepted some but not all of the responses. A response earns reward 1 when the training judge, Gemma-4-12B-it (Gemma Team et al., 2026), finds every essential criterion met, every pitfall avoided and the weighted share of met criteria at least 0.5, and reward 0 otherwise, including when the judge's output cannot be parsed. The judge reasons for at most 400 tokens and then scores every criterion; it is sampled at temperature 0.6 with at most 2,400 tokens. Responses are capped at 2,048 tokens. Every run in the main text uses the training recipe with the overlong penalty, which adds DAPO's soft overlong penalty (Yu et al., 2025) to the judge's reward before group normalization (zero up to 1,536 tokens, falling linearly to -1 at 2,048); runs without the overlong penalty are reported in Appendix F.6. MMLU-Pro STEM (Wang et al., 2024) consists of the 3,148 physics, chemistry and biology questions of the MMLU-Pro test set, asked zero-shot with an instruction to end with the chosen letter, decoded greedily with 2,048 tokens and scored every 50 steps; the answer is read from the final "Answer:" line, with a boxed option as a fallback. GPQA-Diamond (Rein et al., 2024) (198 questions) is evaluated for the Qwen3-8B-Base runs with the same protocol.

Judges in the judge-strength comparison and in calibration. The judge-strength runs change only the training judge. The stronger judge, Gemma-4-26B-A4B-it (25.2B parameters, 3.8B active) replaces Gemma-4-12B-it with the same prompt, and in the reference-as-reward runs DeepSeek-V4-Flash (DeepSeek-AI, 2026) grades each response with the reference answer in view. The binary pass agreement plotted in Table 2c is measured against a panel of GLM-5.3-Flash (GLM-5 Team et al., 2026) with reasoning and Kimi-K2.6 (Kimi Team et al., 2025) without, on criteria where both agree (Appendix C).

Seeds. Every training run has three seeds. A seed fixes the pool order, the sampler's seed and the selector's random stream, and all methods of a seed share it, which is what makes the per-seed differences paired. Runs are not bitwise reproducible, since the sampler is non-deterministic, so pairing by seed controls the pool order and the random streams but not all run-to-run variation. To measure that variation, GRESO and MoPPS in Math were trained a second time under two of the seeds. The two GRESO runs of a seed differ by 3 and 14 events. The two MoPPS runs differ by 22 and 2. We average the runs of a seed first, so every quantity carries the standard error over the three

seed values, and a difference from uniform sampling carries the standard error of the three per-seed differences. We report mean ± SE and do not run hypothesis tests.

Table 19: Hyperparameters shared by every method (the Qwen3-8B-Base runs use the Science column).

	Math	Science
Policy	Qwen3-1.7B-Base	Qwen3-1.7B-Base, Qwen3-8B-Base
Pool size N; steps	800; 300	406; 200
Prompts per step P; group size G	32; 8	32; 8
Response tokens	$\leq$1,024	$\leq$2,048, overlong penalty from 1,536
Rollout temperature; top-p	1.0; 1.0	1.0; 1.0
Optimizer; β_1, β_2; ϵ	AdamW; 0.9, 0.999; 10^{-8}	same
Learning rate; schedule; warm-up	3×10^{-6}; constant; none	same
Weight decay; gradient-norm clip	0; 1.0	same
Loss; KL penalty	token-level (DAPO); none	same
Clip range $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.2, 0.28)$	same
Optimizer steps per batch	1	1
Precision	FP16 mixed precision	same
Training judge	Gemma-3-12B-it, no reference	Gemma-4-12B-it, rubric
Judge sampling	temp. 0.6, $\leq$2,400 tokens	temp. 0.6, $\leq$2,400 tokens
Evaluation (every 50 steps)	MATH held-out, 1,024 tokens	MMLU-Pro STEM, 2,048 tokens

E.2 Methods and baseline fidelity

Every method uses the same trainer, loss, sampler and judge; only the choice of prompts, and for the weighted methods the weights, differ. Appendix B.3 writes each selection rule as a sampling design.

Dynamic sampling. DAPO's filter (Yu et al., 2025) resamples until the batch holds enough informative groups. We run it as a fixed oversample, which keeps the rollouts per step constant and the comparison with the other methods equal in steps. Each step rolls 48 prompts, trains on up to 32 judge-informative groups, and fills any shortfall with zero-variance groups, whose advantages are zero. A step trains on about 22 informative groups, so our variant is DAPO's filter under a fixed rollout budget rather than a filter that always fills the batch. It therefore generates 1.5 times the rollouts of every other method. The loss normalizer counts the tokens of every rolled response.

GRESO. We ported GRESO from the authors' code (Zheng et al., 2025a) rather than from the paper. A prompt whose last groups were all correct (mean reward at least 0.98) or all wrong (at most 0.11) accumulates a streak $c_{t,i}$ and is kept with probability $\max(\phi^{c_{t,i}}, 0.01)$, with separate base rates ϕ_{easy} and ϕ_{hard}. Both start at 0.5. After the first epoch they move by 0.01 per step within $[0.05, 0.95]$. The target is a zero-variance share of 0.25 in the trained batch, one third easy and two thirds hard. We changed three things, none of which touches the rule. The zero-variance verdict comes from our reward pipeline rather than being recomputed, an epoch is defined as tP/N steps, and candidates come from a cycling stream of the pool without repeats within a step, with any shortfall filled uniformly and counted.

MoPPS. We use the authors' code for MoPPS (Qu et al., 2026a) unchanged at a fixed commit. Each prompt carries a Beta$(1, 1)$ prior on its pass rate, updated after each visit by G times the group's mean reward and G times its complement, without forgetting. Each step draws a Thompson sample for every candidate and keeps the 32 whose samples are closest to 0.5. The candidate set is the authors' $16P = 512$ prompts, capped at the pool size, which is the whole pool in Science.

Importance-weighted selection. IW-GRESO takes GRESO's keep probability (the product of its easy and hard keep terms) as $\kappa_{t,i}$, and IW-MoPPS takes the posterior mass of the prompt's pass rate in $[0.3, 0.7]$; both are floored at 0.3. Inclusion probabilities $\rho_{t,i} = \min(1, \lambda\kappa_{t,i})$ with $\sum_i \rho_{t,i} = P$ are found by water-filling, and exactly P prompts are drawn by systematic probability-proportional-to-size sampling over a fresh random permutation of the pool, on a random stream separate from the selector's. Each group's advantages are multiplied by $P/(N\rho_{t,i})$ after group normalization, and the

loss normalizer is fixed before the draw as an exponential moving average (rate 0.2) of past weighted token counts. Algorithm 1 gives the pseudocode. For MoPPS the draw replaces its top-k step; the selector's state is updated as usual.

DPPO. We follow DPPO's prompt pruning (Zhu et al., 2026a) with its default rates $r_q = r_o = 0.9$: among a uniformly drawn batch of 32, the half with the lowest mean absolute advantage at their previous visit are candidates, 90% of the candidates are pruned, and the kept candidates are reweighted by the inverse of their keep probability; completions are pruned at rate r_o in the same way. As in DPPO, the first epoch is not pruned and pruned prompts are not replaced, so a step trains on 18 groups in steady state and a run uses 0.59 (Science) and 0.60 (Math) of uniform sampling's rollouts. DPPO was designed for rule-based rewards; we apply it unchanged under the judge.

Prompt Replay. We follow Baroian & Berger (2026) with replay probability 0.75 and a cooldown of 10 steps after each use. A prompt is replayed at most 15 times per stay in the buffer, and it enters the buffer when its pass rate lies in $[0.25, 0.75]$. The cooldown and the reuse bound limit its visit amplification to 1.90 in Science and 2.50 in Math.

Noise-corrected GRPO. We apply the GRPO variant of El Mansouri et al. (2026) on top of GRESO. It debiases each reward affinely with one global pair of false-positive and false-negative rates estimated on the judge's calibration set (Appendix C). The pair is 0.213 and 0.163 for the Science judge. For the Math judge it is 0.222 and 0.163. The method normalizes the advantages by a noise-corrected standard deviation instead of the group's. The affine correction rescales the response-dependent error term by $1/(1-\alpha-\beta)$ rather than removing it, and it does not change the zero-variance verdicts GRESO reads, so GRESO's selection rule is unchanged; Table 40 reports its visit amplification as measured.

Safeguards. The visit cap skips a prompt once it has at least 8 visits and more than $\Lambda_{\max} \max(1, tP/N)$ of them, and it fills the freed slots uniformly. We use $\Lambda_{\max} = 1.5$ or 2.1. The cooldown makes a prompt ineligible for 12 steps after each visit. The visit-triggered audit starts when a prompt's visits exceed 1.5 times uniform sampling's share in Math and 2.1 times in the Science audit runs (Appendix F.18). A visit counts as exploited when the judge accepts at least one response and the auditor none; two consecutive exploited visits retire the prompt, and one visit on which the auditor also accepts a response exempts it until its visit count doubles. The auditor is the exact verifier, or GLM-5.3-Flash as the auditing judge (temperature 0.6, at most 2,400 tokens).

Selectors we do not run. GradAlign (Yang et al., 2026b) selects training problems whose policy gradients align with those of a small trusted validation set and is evaluated under unreliable rewards, and VCRL (Jiang et al., 2025) schedules prompts by the variance of their group rewards. VCRL reads the same judged rewards as GRESO and MoPPS, so our analysis applies to it; GradAlign needs a trusted validation set, which our correction does not, and the two can be combined, since the correction wraps any selector's keep propensities.

E.3 Compute

Every training run used one node with eight NVIDIA A100 GPUs, on which the policy trainer, the sampler and the training judge were co-located (the judge served by two tensor-parallel inference engines), so the judge's cost is included in the GPU-hours below, except in the reference-as-reward runs, whose reward came from DeepSeek-V4-Flash through an API. We count GPU-hours as billed node-hours times eight. One 1.7B arm-run costs about 19 GPU-hours with a judge in the loop, a Math arm-run with the exact verifier as the reward about 11 GPU-hours, and one Qwen3-8B-Base arm-run about 99 GPU-hours. Table 20 lists the compute by group of runs; each group includes all three seeds of every arm in it, and the Math selector groups include the second runs at seeds 0 and 1 (Appendix J).

The development figure covers every other job we started, among them about 110 GPU-hours of jobs that were stopped or failed before producing a result; none of those jobs contributes to a reported number. Two further items are not counted in GPU-hours. First, the offline re-evaluation of saved checkpoints (MMLU-Pro STEM, held-out MATH and held-out rubric responses) and the base-policy responses used to build the Science pool and to calibrate the judges ran on separate

Table 20: Compute ledger in A100 GPU-hours, rounded to the nearest ten, with the number of training runs in each group. Judge serving is included except for the API-served reward of the reference-as-reward runs.

Group of runs	Reported in	Runs	GPU-hours
Math (1.7B), judge and exact-verifier arms, replay controls, visit cap, cooldown and audit arms	Sec. 3, 6; App. F	–	1,850
Science (1.7B), with the overlong penalty, main arms and judge-strength runs	Sec. 3, 6	–	990
Science (8B), main arms	Sec. 6	27	2,660
Runs reported in the main text			5,500
Science (1.7B), without the overlong penalty	App. F.6, F.18	18	340
Floor sweep, IW-GRESO and IW-MoPPS at floors 0.1 and 0.5, both domains	App. D.13	24	460
Floored GRESO without weights, both domains	App. F	6	110
Math pool size (400 and 1,600 prompts, uniform sampling)	App. F	6	110
Math pool of 3,200 prompts (uniform, GRESO, IW-GRESO, GRESO with bound and verification)	App. F	12	400
Math (8B), uniform, GRESO, IW-GRESO, GRESO with bound and verification	App. F	12	1,000
Math gradient masking (GRESO and uniform sampling)	App. F.6	6	110
Group size, Science (1.7B, $G = 16$ and $G = 32$)	App. F.15	24	1,210
Group size, Math (1.7B, $G = 16$)	App. F.15	9	310
Learning rate, both domains (1.7B)	App. F.16	24	460
Unfiltered Science pool (1.7B)	App. F.17	12	230
Runs reported only in the appendix			4,740
All reported runs			**10,240**
Development (pilots, smoke tests, judge pools, learning-rate bracket, stopped jobs)			300

single-GPU instances, which together amount to a small fraction of the training compute. Second, hosted models were called through an API. The largest item is the reward of the 12 reference-as-reward runs, about 0.7 million DeepSeek-V4-Flash verdicts (three seeds, each with three arms of 200×256 responses and dynamic sampling with 200×384). The rest are the two panel models for the Science calibration (about 3,040 calls), DeepSeek-V4-Flash as the reference judge in calibration and as the auditing judge of the Science visit-triggered audit, GLM-5.3-Flash as the auditing judge in one variant of the Math audit, and both panel models for the offline scoring of held-out rubric responses.

F Additional results

F.1 Per-seed values of the diagnosis

This subsection lists every Section 3 quantity with its per-seed values. In Math, seeds 0 and 1 ran each selector twice, so for those seeds the tables give both runs and a seed's value is the mean of its runs. The point estimate is the mean of the three seed values, every SE is computed over them, and differences from uniform sampling are paired within each seed. Same-seed runs differ more than seeds for event counts, the gap and the spurious-group share, which is why both runs are listed. The first runs of the Math selectors in these two seeds stopped early, between steps 274 and 288 (Appendix J). Quantities computed over the steps a run completed use both runs, and held-out accuracy at the end uses the second run. Accuracies, mastery values and shares are in percent and differences in points. We report mean $\pm$ SE and no significance tests. Tables 31 and 32 give the per-seed values of every further quantity quoted in this appendix.

For MoPPS the table gives per-seed means of visit amplification; its runs range from 12.63 to 13.42. The chance expectation of Corollary 1 (Appendix D.6) sums the probability of an event under response-independent error over each run's own at-risk visits, using each prompt's false-positive

rate on its earlier all-wrong groups. Events beyond chance subtract it, and the two Math metrics then tell one story. The selectors create about three times as many at-risk visits as uniform sampling. They produce events on these visits at about uniform sampling's rate, 7.2 and 7.6 per 1,000 against 7.7. Chance expects a lower rate on their visits, 4.6 and 3.2 per 1,000 against 7.4, because most of them fall on prompts the judge rarely over-credited before. The difference, 10.8 and 18.5 events against 0.6, is what response-dependent error adds to the exposure, and much of it accrues beyond the twelfth visit (Appendix F.3).

Held-out MATH accuracy does not separate the arms. Both selectors are below uniform sampling in seed 1 and above it in seed 0, so the paired differences change sign across seeds, which is why Section 3 reports no consistent change. With the exact verifier as the reward, the selectors' held-out accuracy differs from uniform sampling's by $+0.1 \pm 1.7$ (GRESO) and $+1.4 \pm 1.4$ points (MoPPS). From the exact-verifier runs to the judge runs, GRESO loses 9.7 ± 1.1 points of its lead in strict mastery. Same-seed GRESO runs differ by 1.4 to 2.8 points, so this loss lies well outside run-to-run variation.

Replay control. Table 22 gives the per-seed values behind Table 1. The replay in each seed replays the schedule of one GRESO run of that seed, its source run, so the source runs have the same visit amplification and error-prone visit share as their replays. The difficulty-matched replay permutes prompt identities within groups of equal base pass count, which keeps every visit count and the per-step difficulty profile. In seed 0 the replay shares 13 of its 26 events with its source run, and in the other seeds about half of its events as well (Table 31). The replay exceeds uniform sampling by $+15.3 \pm 2.0$ events. The matched replay is not free, since training on the same schedule over other prompts leaves it below uniform sampling in strict mastery (67.5% against 72.8%).

Across the 17 Math arms of the study, the Spearman correlation of reward-hacking events is 0.72 ± 0.14 with visit amplification and 0.78 ± 0.11 with error-prone visit share (SE from the Fisher z-transform). Neither coefficient identifies a cause, and the replay pair, which holds amplification fixed, is the contrast that separates the two quantities.

How $\hat{\mathcal{E}}$ was identified. The error-prone set holds 106 prompts, 90 of them in $\mathcal{H}$. A prompt enters it when its judged count exceeds its verified count ($k^{\mathrm{obs}} > k^{\mathrm{true}}$) on at least half of its visits, with at least four visits, in the uniform runs of seeds 0 and 1. It is identified on those two uniform runs only, so it is independent of every selector, replay and weighted run it is used to evaluate. The uniform share (13.3%) is fixed by design, since uniform sampling visits every prompt equally often. The error-prone visit share and the event count are related by construction, since both follow over-crediting, which is why the main text reads the replay contrast as showing the role of the repeated prompts rather than of this share as such.

Science. Table 23 lists MMLU-Pro STEM (3,148 questions, 2,048 generated tokens) at every evaluation for every arm, and Table 24 gives the per-seed values at the saved checkpoints. The base policy scores 18.3% before training in every arm. GRESO is already 3.5 ± 0.9 points behind at step 25 and then stays roughly flat. MoPPS matches uniform sampling through step 50, then declines at every checkpoint to 37.9%, and is 7.8 ± 1.0 points behind at step 150; its share of style-split groups keeps rising over the same steps (Appendix F.2). No selector checkpoint reaches uniform sampling's final $48.9 \pm 0.4\%$.

At step 50 GRESO's visit amplification is 1.78 in every seed. Its most-visited prompt then has 7 visits, as many as dynamic sampling's, against the 2.67 expected under an i.i.d. uniform draw of the same size; uniform sampling's is 1.08 over the whole run. Dynamic sampling shows no loss at step 50, and MoPPS shows none at a visit amplification of 6.1, so visit amplification alone does not account for GRESO's early deficit of 4.3 ± 1.1 points. Since visit amplification is a maximum over prompts, it cannot exclude a broader concentration of visits; the style-split share of Appendix F.2 measures that composition directly.

At step 200, pooled over the two selectors, the deficit is 10.6 ± 0.6 points. Each selector is at least 5 points below uniform sampling in all three seeds, and at least 8 points below in two seeds (GRESO) and three seeds (MoPPS).

Formatting. We call an answer reasoned if it has at least 150 tokens and terse otherwise. In seed 0, 56% to 63% of the selectors' answers are terse, against 11% of uniform sampling's. We then

compare accuracy on the items where both arms reason (1,169 items for GRESO against uniform sampling in seed 0). There GRESO is 13.0 ± 3.0 points less accurate than uniform sampling and MoPPS 13.2 ± 0.6 points. The loss therefore persists when the selectors reason at length.

Held-out rubric prompts and in-domain mathematics. We re-grade the completions of every run on 300 held-out RaR-Science prompts with one fixed stronger judge, Gemma-4-26B-A4B-it, using the training rubric and reward rule, so that only the judge changes. Its pass rate at the end of training differs from uniform sampling's $41.7 \pm 0.6\%$ by -1.6 ± 1.0 points for GRESO and -1.0 ± 0.4 for MoPPS. Dynamic sampling is 5.1 ± 0.8 points lower. The rate at which the training judge passes a completion that the stronger judge fails, the rate at which the two judges disagree in the training judge's favor, is $39.5 \pm 1.6\%$ for MoPPS against $32.4 \pm 0.7\%$ for uniform sampling. MoPPS therefore exploits the rubric more ($+7.1 \pm 1.0$ points; GRESO $+2.2 \pm 1.3$) although its pass rate under the stronger judge is only slightly lower. On 500 held-out MATH prompts at 1,024 generated tokens the Science runs show no in-domain regression, since the selectors score $86.0 \pm 0.4\%$ and $85.3 \pm 0.2\%$ against $83.9 \pm 0.4\%$ for uniform sampling.

F.2 The Science mechanism: style-split groups, terse answers and derivation items

Style-split groups. A trained Science group is style-split when the training judge splits it ($0 < k^{\text{obs}} < G$) while the reference judge, DeepSeek-V4-Flash grading the same responses against the reference answer, gives all G responses the same verdict (Section 2). Such a split does not track correctness and follows surface features of the responses. The weak rubric judge rewards such features often, since it passes 82% of responses that only restate the question (Appendix C). That rate is comparable to the rates prior work reports for open-weight LLM judges on content-free responses (Zhao et al., 2025), and it agrees with the exploitation of rubric judges reported by Mahmoud et al. (2026). We grade 64 trained groups per 25-step window and run with the reference judge.

Persistence enriches for style. A split that tracks correctness resolves as the policy learns the prompt, while a style-driven split does not. Under uniform sampling, the style-split share of judge-informative groups rises with the number of consecutive judge-informative visits just before. It is $15.1 \pm 0.3\%$ on a prompt's first informative visit, $21.9 \pm 0.6\%$ after one and $31.7 \pm 1.0\%$ after two or more. A history-based selector keeps exactly the prompts whose groups were persistently split, so it enriches its batches for style-split groups without concentrating visits on individual prompts. Dynamic sampling selects on the current step's split without history, so it does not enrich for persistence.

Composition from the first steps. Table 25 shows that GRESO's batches carry more style-split groups than uniform sampling's from the first window, while MoPPS's share rises at every window. The enrichment changes the composition of each batch, so it needs no prompt to be visited more often than under dynamic sampling. This is why GRESO falls behind by step 25 while its visit amplification at step 50 is still 1.8. Importance-weighted selection makes the expected update equal to uniform sampling's, and its share stays within about two points of uniform sampling's.

The learned style. The policy learns what the judge rewards. Table 26 gives the share of terse MMLU-Pro answers over training. It rises in the selector arms from about step 25, to $48.7 \pm 9.4\%$ for GRESO and $59.3 \pm 2.3\%$ for MoPPS at step 200. Uniform sampling, dynamic sampling and the importance-weighted arms stay within a few points of the base policy's 9%. GRESO's milder seed 1 matches its smaller accuracy loss in that seed. At step 200 the mean response length on MMLU-Pro is 514 ± 10 tokens for uniform sampling, against 370 ± 41 for GRESO and 318 ± 10 for MoPPS, while the importance-weighted arms stay near 480 to 490 tokens. In training responses of steps 150 to 199, GRESO and MoPPS restate rubric terms in $42.3 \pm 3.3\%$ and $52.0 \pm 1.7\%$ of responses, against $21.3 \pm 0.9\%$ for uniform sampling.

Where the accuracy is lost. MMLU-Pro STEM rewards derivation, and the terse style skips it. The reference panel labels 1,812 of the 3,148 items as requiring a multi-step derivation or calculation. On these items GRESO is 15.8 ± 3.2 points below uniform sampling and MoPPS 17.0 ± 1.4 points. On the remaining items the gaps are only 2.4 ± 0.7 and 3.0 ± 0.4 points. Weighted by the item counts, the two parts combine to each seed's overall deficit.

Consistency checks. Four observations agree with this account. The loss shrinks with the stronger judge (Section 6), which splits fewer groups on style, and it is not detectable with the reference judge as the reward, which grades against the reference answer. Importance-weighted selection restores uniform sampling's share of style-split groups (Table 25) and recovers most of the loss. On held-out rubric prompts the pass rates under the stronger judge stay similar across arms while MoPPS passes the training judge alone more often, so the learned style shows up as exploitation on the training task and as lost accuracy out of distribution. Finally, the Science selectors score slightly higher than uniform sampling on held-out MATH, whose short numeric answers the terse style does not penalize.

F.3 Number of prior visits and the rate of reward hacking

The judged-minus-true gap in Figure 2b grows with the visit number over the first 48 visits. Table 27 shows the same effect as a rate. A visit n to a prompt of $\mathcal{H}$ is at risk if visit $n-1$ was judge-informative with at least one verified correct response and the prompt has had no reward-hacking event yet (Section 2). The event rate is the share of at-risk visits at which the event occurs. The strict-mastery rate is the share at which the prompt reaches $k^{\text{true}} = G$, excluding visits after an event. Uniform sampling never visits a prompt more than 12 times.

In every selector run the event rate is higher at visits 13 to 24 than at visits 1 to 12, while the strict-mastery rate falls by a factor of three or more. The prompts that remain informative after a dozen visits are therefore increasingly those the judge over-credits. In the same visit ranges, the count of events per strict mastery rises from below 0.1 at the first 12 visits to between 0.2 and 1.75 later. Pooling GRESO and MoPPS, Table 28 splits the event rate by the number of steps since the previous visit. The rise with visit number appears at every interval between visits, so the number of prior visits, not the interval between them, carries the effect.

Visits needed for mastery and visits at which events occur. Under uniform sampling, $93.3 \pm 0.3\%$ of the prompts strictly mastered by the end of training are mastered within their first six visits. The selectors' reward-hacking events come later, since $80.2 \pm 0.6\%$ of them occur after a prompt's sixth visit, with per-run shares from 69% to 87%. After the twelfth visit, which uniform sampling never exceeds, $64.3 \pm 0.7\%$ of the selectors' events still occur. Most of the selectors' events therefore occur at visit counts beyond those at which uniform sampling masters most prompts. The share of the selectors' at-risk visits that fall after the sixth visit is not reported separately; the event rates of Table 27, which are per at-risk visit, carry that comparison.

F.4 Overlap of prompts with reward-hacking events under two definitions

The main text uses the event definition, under which a prompt counts for a run if it has a reward-hacking event on $\mathcal{H}$. A looser definition counts a prompt if any trained visit, anywhere in the pool, has no verified correct response while the judge accepts at least 6 of 8. Table 29 gives the Jaccard index between the sets of two runs under both definitions, with the overlap of random sets of the same sizes (20,000 draws).

The three uniform pairs cover every pair of seeds. In the pair of seeds 0 and 1 each run has 11 events, and they share 4. Random sets of these sizes in $\mathcal{H}$ reach their Jaccard index of 0.22 with probability about 10^{-4}. In these two seeds, 14 of the 18 prompts with an event under uniform sampling also have one in at least one selector run. Under both definitions every pair of runs overlaps far more than random sets of the same sizes. The difficulty-matched null draws, for each run, a random set of the same size from the same strata of base pass count as that run's event prompts (20,000 draws per pair; Monte Carlo SE below 0.002). Hard prompts do recur for difficulty alone, which raises the null from 0.02–0.04 to 0.09–0.13, but every observed pair stays about twice as high or more. The recurrence therefore exceeds what difficulty explains, as the main text concludes. It does not separate the judge from other prompt-specific properties that the difficulty strata miss.

F.5 All-rejected solvable groups, rollouts per informative group and time to mastery

All-rejected solvable groups. These are trained groups in which the judge rejects all G responses although the verifier accepts at least one ($k^{\text{obs}} = 0 < k^{\text{true}}$). Their share is a lower bound on judge false negatives, since it misses groups in which only some correct responses are rejected. They make up $3.73 \pm 0.22\%$ of GRESO's groups, against $0.60 \pm 0.05\%$ for uniform sampling and $0.38 \pm 0.02\%$ for MoPPS. Although these groups look zero-variance, GRESO does not skip their prompts for long, since it revisits them after a median of 2 steps in every seed (MoPPS after 50 and uniform sampling after 28).

Rollouts per truly informative group. Over the last 100 steps, GRESO and MoPPS spend 17.2 ± 0.2 and 17.3 ± 0.1 rollouts per truly informative group, against 36.1 ± 0.6 for uniform sampling. Over the whole run the values are 15.1 ± 0.1, 14.5 ± 0.2 and 27.1 ± 0.5. Under the judge the selectors therefore still find truly informative groups at about twice uniform sampling's rate, consistent with their remaining lead in strict mastery.

Time to mastery. Table 30 gives the first step at which 80% of $\mathcal{H}$ is strictly mastered, under the judge and under the exact verifier as the reward, for the same seed. Runs that do not reach 80% are censored at their last step, so their ratios are lower bounds, and we report ranges with per-run values instead of a mean. With the exact verifier as the reward, GRESO and MoPPS cross at steps 86.3 ± 1.7 and 89.7 ± 3.8, against 163 for uniform sampling in every seed.

F.6 Training without the overlong penalty, answer extraction and gradient masking

Science without the overlong penalty. Our first Science runs used the training recipe without the overlong penalty, the training library's defaults. Under it the policy's completions lengthen until most of them reach the 2,048-token limit, and selection brings this runaway growth of response length forward. We define its onset as the first step after which more than half of the step's completions are truncated for 10 consecutive steps. The onset comes at steps 88.3 ± 9.0 (GRESO) and 86.0 ± 5.2 (MoPPS), against 150.7 ± 6.1 for uniform sampling. At step 200 on MMLU-Pro STEM, GRESO and MoPPS are then 10.3 ± 4.5 and 24.4 ± 2.3 points below uniform sampling's $49.6 \pm 0.3\%$. These values use the earlier answer extractor, which changes them by at most 0.3 points. In seed 0 the runaway growth adds 7 points to GRESO's loss and 17 to MoPPS's, relative to the recipe with the overlong penalty. GRESO's loss varies from 3 to 18 points across seeds because the onset varies, so its SE is large. The safeguards evaluated under this recipe are reported with the other ablations of this appendix.

Answer extraction. MMLU-Pro answers are extracted from the final boxed expression. Our earlier extractor accepted only an option letter, and dynamic sampling, more often than the other arms, boxes the text of the chosen option instead. Under that extractor dynamic sampling scored $39.1 \pm 0.3\%$, while the corrected extractor, which also matches an option's text, gives $49.0 \pm 0.3\%$. Every Science accuracy in the paper uses the corrected extractor unless stated.

Gradient masking in Math. Masking the gradient of truncated completions does not change the diagnosis. With masking, GRESO produces 26.3 ± 2.6 reward-hacking events on $\mathcal{H}$ and uniform sampling 11.0 ± 0.6, a difference of 15.3 ± 2.0 events. Their strict mastery at step 200 is $77.0 \pm 0.3\%$ and $72.8 \pm 0.3\%$. Both lie within the range of the unmasked runs.

F.7 Per-seed values of the main results

Tables 33–35 give the per-seed values behind Table 2. Within a seed, every method shares the pool order and the random streams of that seed, so the difference from uniform sampling is paired; the mean $\pm$ SE of the main tables is taken over these three paired differences.

The exact-verifier mastery of each row comes from a run with the same seed and method in which the exact verifier assigns the reward (the exact-verifier run); the judge mastery comes from the judge-rewarded run. Both are graded by the exact verifier. We read mastery at step 200 because the

selectors' exact-verifier runs saturate afterwards (GRESO's reaches 98% by then). Under the exact verifier noise-corrected GRPO estimates flip rates of zero and reduces to GRESO, so its verifier-reward mastery reuses GRESO's exact-verifier runs.

F.8 Safeguards and ablations

Table 36 collects the Math safeguards and ablations, all trained with the Gemma-3-12B-it judge for 300 steps with three seeds. Run-to-run variation matters for reading them. Two runs of GRESO with the same seed differ by 3 and 14 events; two runs of MoPPS differ by 22 and 2. The table gives levels; paired differences are in the text.

Visit bounds. A visit cap at 1.5 times uniform sampling's share lowers GRESO's gap by 0.057 ± 0.006, about 40%, and halves its events. Mastery under the judge and held-out accuracy do not change. Part of the cap's effect is lost to the uniform fills it requires: once the cap binds, 9 to 11 of a step's 32 prompts are forced uniform fills. The arm with the larger cap fills $2{,}578 \pm 19$ prompts per run in this way, and the cooldown arm $3{,}357 \pm 27$. At equal maximum visits, the cap and the cooldown differ by -2.0 ± 0.6 events for GRESO and $+7.0 \pm 2.3$ for MoPPS. Both differences lie inside the run-to-run variation, which is the reading our analysis plan wrote down beforehand. The cooldown and the audit do not compose, since together they give 19.3 events, more than the audit alone.

Visit-triggered audit. With the exact verifier as auditor, the audit brings GRESO's gap to 0.012 ± 0.001, which is 0.028 ± 0.003 below uniform sampling's. It costs 416.0 ± 2.3 audits per run, 4.3% of the trained groups, and retires 29.7 ± 0.3 prompts. Its events (17.3 ± 1.5) stay above uniform sampling's, since a prompt is audited only after it crosses the cap and an event can precede the audit. The audit does not bound visits, so the audited arms keep GRESO's and MoPPS's visit amplification. With GLM-5.3-Flash as the auditing judge in place of the verifier, 15.0 ± 0.6 of the 23.7 ± 0.9 retirements are justified by the exact verifier. The gap is then 0.037 ± 0.002, and the auditing judge costs about one US dollar per run.

Floored draw without weights. IW-GRESO changes two things at once. It softens GRESO's selection into a proportional draw floored at 0.3, and it reweights the drawn groups. To separate them we train the same floored draw with every weight set to 1, in both domains. Its weighted exposure then equals its realized visit amplification, which is close to IW-GRESO's, so Proposition 2 bounds its bias as that of an arm with about threefold amplification, and its update keeps part of GRESO's change of objective. In Math it produces 20.3 ± 1.2 events, 9.5 ± 1.3 fewer than GRESO but 8.3 ± 0.7 more than IW-GRESO. Its gap, 0.071 ± 0.003, is at the level of the visit caps, and its strict mastery under the judge lies between GRESO's and IW-GRESO's. In Science it ends 4.6 ± 0.9 points below uniform sampling and 3.8 ± 0.2 below IW-GRESO, which is 5.5 ± 1.5 points above GRESO. Softer selection alone therefore removes about half of GRESO's excess events and half of its Science loss, and the weights remove the rest.

Pool size. Under uniform sampling, a larger pool means fewer visits per prompt at the same number of steps: halving the pool doubles the visits and doubling it halves them. Normalized to the 800-prompt pool's hard set, events fall from 18.6 ± 2.9 at 400 prompts to 7.0 ± 0.9 at 1,600 (Table 36). The difference at the smaller pool is within about two standard errors, so we read the result as a trend consistent with Proposition 3. Because the hard set grows with the pool, the raw counts on each pool's own hard set reverse the order, and normalizing the smaller hard set doubles its Poisson spread. Strict mastery falls with the pool for the same reason: each prompt is seen fewer times by step 200.

F.9 Chance-adjusted events for every arm

Table 37 gives, for every Math arm, the reward-hacking events on $\mathcal{H}$, the number expected by chance under response-independent errors (Corollary 1, computed per run on the run's own at-risk visits; Appendix D), and their difference. For weighted arms the at-risk visits are weighted. Arms that do not amplify visits sit near uniform sampling's small excess, the visit bound alone slightly above it, and the arms that amplify without weights clearly above. The event rate per 1,000 at-risk visits is

similar across uniform sampling, GRESO and MoPPS (Appendix F.3), so the selectors' excess of events comes from their larger number of at-risk visits; the chance-adjusted count separates the part that chance does not explain.

F.10 A POOL OF 3,200 PROMPTS

The main Math runs revisit each prompt about 12 times under uniform sampling. To test a setting closer to the pool sizes GRESO was designed for, we train with a pool of 3,200 MATH prompts drawn by the same admission rule. With the same steps, P and G, uniform sampling visits each prompt about three times. The hard set is again the pool's prompts that the base policy solves at most twice in eight samples. Table 38 reports uniform sampling, GRESO, IW-GRESO and GRESO with the visit bound and triggered verification, three seeds each. GRESO still produces more reward-hacking events than chance allows, although the excess is smaller than with 800 prompts, and it still gives up about half of its exact-verifier lead in strict mastery under the judge. Both mitigations bring the excess back to uniform sampling's level; the triggered verification costs 2.1% more verifier calls, since fewer prompts exceed the bound in a larger pool. Held-out MATH shows no consistent difference.

F.11 QWEN3-8B-BASE IN MATH

We repeat the core Math comparison with Qwen3-8B-Base, keeping the 800-prompt pool, the Gemma-3-12B-it judge and the hyperparameters of the 1.7B runs (Table 39). The stronger policy produces fewer reward-hacking events under every method, but GRESO's excess over chance remains several times uniform sampling's, and its exact-verifier lead in strict mastery again shrinks under the judge. Both mitigations remove the excess; the triggered verification costs 3.6% more verifier calls. Held-out MATH shows no consistent difference.

F.12 JUDGE STRENGTH

Table 2c gives the final MMLU-Pro STEM differences for each training judge. The judge is the only thing that changes between the three sets of runs; each set has three seeds of uniform sampling, dynamic sampling, GRESO and MoPPS, and dynamic sampling stays within 0.3 points of uniform sampling under every judge. The agreement of a judge is its binary pass agreement with a panel of GLM-5.3-Flash with reasoning and Kimi-K2.6 without, on responses where the two agree (1,472, 1,519 and 1,519 calibration responses). Its binomial standard errors ignore the clustering of eight responses per prompt and are therefore lower bounds. The third judge differs in kind, since it grades against the reference answer. The analysis plan classified the loss as judge-driven if the deficit under the stronger judge was within 5 points of uniform sampling, and both selectors fall in that band. The training false-positive rates on the same set are 0.213 ± 0.011, 0.021 ± 0.004 and 0.008 ± 0.002. The change from the 12B judge to Gemma-4-26B-A4B-it is not explained by parameter count alone, since the 26B judge has only 3.8B active parameters, in line with the gap between weak and strong rubric judges reported by Mahmoud et al. (2026).

F.13 QWEN3-8B-BASE

The Qwen3-8B-Base block uses the Science pool, judge, recipe and hyperparameters of the 1.7B runs, with three seeds per method (Table 34). Uniform sampling reaches 68.6 ± 0.5 on MMLU-Pro STEM. The selectors' loss is smaller than at 1.7B (-6.9 ± 0.6 for GRESO and -8.2 ± 0.6 for MoPPS) but appears in every seed. The weighted selectors again end within one point of uniform sampling, IW-GRESO at -0.6 ± 0.7 and IW-MoPPS at -0.9 ± 0.6. On GPQA-Diamond, uniform sampling answers $43.4 \pm 1.3\%$ of the 198 questions. Every selector moves in the same direction as on MMLU-Pro STEM, by less than two standard errors; GRESO's difference, the largest, is -3.2 ± 1.9 points (per-seed counts in Table 34). With 198 questions, the per-seed paired differences have a standard deviation of about 3 points, so three seeds cannot resolve differences of this size. Table 41 gives the compute measurements of the block. At this size noise-corrected GRESO's visit amplification is 7.36 ± 0.36 and Prompt Replay's 1.94 ± 0.04.

F.14 Additional measurements of the methods

Further baselines. Table 40 repeats the three further baselines of Table 2 with uniform sampling for reference and the mastery levels behind their Math differences. Noise-corrected GRPO applied to GRESO behaves like GRESO in both domains, because its affine correction with response-independent flip rates rescales the response-dependent error term by $1/(1-\alpha-\beta)$ instead of removing it and leaves GRESO's selection rule unchanged (Corollary 1); its visit amplification, measured on its own runs, is close to GRESO's but not equal, and its verifier-reward mastery reuses GRESO's exact-verifier runs. DPPO weights its kept prompts by their inverse keep probability, so its expected update equals uniform sampling's, and it stays near uniform sampling on every measure with about 0.6 of its rollouts; its effective sample size is lower because of rare weight-10 groups. Prompt Replay's cooldown and reuse bound limit its visit amplification, and it falls between uniform sampling and the selectors. Its Math events exceed uniform sampling's by 7.7 ± 1.8, and its excess over chance is 4.1 ± 0.5.

Table 41 gives their Science compute measurements. In Math, held-out MATH differs from uniform sampling by less than one point for all three, and their judged-minus-true gaps follow the events (Table 42).

Compute and exposure in Science. Table 41 separates the two quantities that Table 2a merges for the weighted rows, the number of judge-informative groups per 1,000 rollouts and their Kish effective sample size. The weighted selectors still draw more informative groups than uniform sampling, but their weights offset this advantage and leave an effective sample size 5% below uniform sampling's. Their largest realized weighted exposure stays close to what i.i.d. uniform draws of the same size produce, whereas DPPO's rare weight-10 draws push its own well above it.

Accuracy where both runs reason. On the MMLU-Pro STEM questions where both the method's response and uniform sampling's are at least 150 tokens long, the selectors' loss persists, so it is not only a shift toward terse answers. With Qwen3-1.7B-Base, GRESO answers $33.4 \pm 3.3\%$ of these questions against 46.4% for uniform sampling. MoPPS answers $32.4 \pm 0.4\%$ against 45.6%. The weighted selectors, dynamic sampling and DPPO stay within two points of uniform sampling on their subsets. The comparison set differs for each pair, which is why uniform sampling's value changes.

Exposure and gaps in Math. Table 42 collects the Math exposure measurements. The weighted share of exposure to the error-prone set $\hat{\mathcal{E}}$ is the same for the weighted selectors as for uniform sampling, while their unweighted visit share is higher. The weighted rows therefore still visit error-prone prompts more often, at a proportionally smaller weight. Their visit amplification stays far below the unweighted selectors', and their largest realized weighted exposure stays close to that of i.i.d. uniform draws. We report the gap of weighted rows weighted by the importance weights, which is the gap of the expected update. DPPO's Kish effective sample size depends on a few weight-10 groups.

Held-out suite in Math. Table 43 extends the Math evaluation to AMC 2023, Minerva Math and OlympiadBench. No method differs from uniform sampling by more than about two standard errors on any benchmark, consistent with held-out MATH.

F.15 Group size

The main experiments use $G = 8$. We repeat the core comparison with $G = 16$ and $G = 32$ in Science and with $G = 16$ in Math, keeping $P = 32$ prompts per step, the pool, the judge, the learning rate and the number of steps fixed, so that each step rolls $32G$ responses (dynamic sampling rolls $48G$). The comparison is therefore at an equal number of training steps, as in the main text, and a larger group spends proportionally more rollouts per step. Every cell has three seeds. Table 44 reports Science and Table 45 reports Math. Besides the outcome we report three diagnostics of selection. The *visit concentration* is the share of all trained visits that go to the 10% most-visited prompts, which is 0.10 when visits are even (Appendix B.5). The realized visit amplification Λ_{vis} is defined in Section 2. The late zero-variance share is the fraction of trained groups in steps 160–200 whose G judge verdicts are all equal (for dynamic sampling, the fraction of rolled groups).

Science. The transfer loss of the history-based selectors is not specific to $G = 8$. At $G = 16$ GRESO and MoPPS end -9.2 ± 2.0 and -12.3 ± 1.0 points below uniform sampling, while dynamic sampling ties uniform sampling at every group size. The two selectors respond to the group size in opposite directions, and the diagnostics show why. MoPPS adds G pseudo-counts to its Beta posterior at every visit, so its posterior sharpens faster with larger groups, its visit concentration rises from 0.451 to 0.517 and its deficit grows from 11.0 to 12.3 and 13.3 points. GRESO skips a prompt only after zero-variance groups, and at $G = 32$ fewer groups are zero-variance (uniform sampling's late zero-variance share falls from 0.570 at $G = 8$ to 0.370), so fewer prompts become eligible for skipping; GRESO's visit concentration falls to 0.209 ± 0.006, its visit amplification to 3.11 ± 0.11, and its deficit to -4.3 ± 1.4 points, with one seed only 1.6 points below uniform sampling. At $G = 32$ GRESO's tuner reaches its bound on the keep rate, so its late zero-variance share stays at 0.11, below its target of 0.25. Uniform sampling itself gains about one point over a fourfold range of group sizes (48.9 ± 0.4, 49.5 ± 0.4, 49.8 ± 0.4), in line with reports that larger groups help modestly at a fixed number of prompts per step (He et al., 2025; Tan et al., 2026) and that 8 to 32 generations per prompt at a fixed total batch leave learning curves nearly unchanged (Khatri et al., 2026).

Math. The selectors produce 2.8 and 3.5 times as many reward-hacking events as uniform sampling at $G = 16$, against 2.6 and 2.9 times at $G = 8$, and their judged-minus-true gap stays three to nine times uniform sampling's. Held-out MATH accuracy shows no consistent difference at either group size (GRESO minus uniform -0.9 ± 2.1 points at $G = 8$ and -0.1 ± 1.7 at $G = 16$; MoPPS -0.5 ± 2.4 and -0.1 ± 1.9; each within about one SE of zero), as in the main text. The stricter event threshold at $G = 16$ does not make chance events more likely. Under a common false-positive rate $\alpha = 0.22$, a visit at risk produces 12 or more accepted responses out of 16 by chance with probability 9×10^{-6}, and under the per-prompt rates of Section 4 (a $\mathrm{Beta}(0.59, 1.74)$ fit) with probability 0.071, against 0.095 for 6 of 8. Proposition 1 implies the direction of the effect, since at $\alpha = 0.22$ a prompt the policy never solves looks judge-informative with probability 0.863 at $G = 8$ and 0.981 at $G = 16$, so larger groups keep the prompts that carry judge errors in the selectors' informative set for longer.

F.16 Learning rate

All main runs use a learning rate of 3×10^{-6} with full fine-tuning. Published GRPO and DAPO recipes at 1.5B to 32B parameters use 10^{-6} with much larger batches (Yu et al., 2025; Liu et al., 2025; He et al., 2025), and ScaleRL uses 5×10^{-7} (Khatri et al., 2026); none reports a learning-rate ablation that we could find. We chose 3×10^{-6} from a short development bracket on Math, in which 10^{-6} learned slowly while 10^{-5} gave no gain in training reward or policy entropy over 3×10^{-6}. Tables 46 and 47 repeat uniform sampling and GRESO at 10^{-6} and 10^{-5}, with $G = 8$ and everything else as in the main runs.

GRESO's excess of reward-hacking events and its judged-minus-true gap grow roughly in proportion to the step size and never change sign. The event ratio to uniform sampling stays between 2.5 and 2.7 at every learning rate, and the gap difference is $+0.058 \pm 0.011$, $+0.089 \pm 0.005$ and $+0.126 \pm 0.022$ at 10^{-6}, 3×10^{-6} and 10^{-5}, and GRESO's visit amplification rises with the step size (8.89 ± 0.07, 11.40 ± 0.10 and 12.61 ± 0.10). GRESO's lead in strict mastery of $\mathcal{H}$ under the judge stays between 3 and 4 points, inside the range reported in Section 3, and held-out MATH shows no consistent difference at any learning rate (-0.3 ± 2.0, -0.9 ± 2.1 and -0.8 ± 2.0 points). In Science the deficit is -10.1 ± 2.1 and -12.4 ± 2.3 points at the two larger learning rates and -5.4 ± 1.6 at 10^{-6}, where every seed is below uniform sampling but the three seeds spread from 2.4 to 7.6 points. At 10^{-6} both arms are undertrained, and uniform sampling at step 200 reaches roughly the accuracy that the main run reaches by step 50. At 10^{-5} uniform sampling itself loses 1.8 points on MMLU-Pro STEM and 2.7 on held-out MATH and varies more across seeds, consistent with faster over-optimization of the judge and with a report that GRPO degrades at this step size in a web-agent setting (Gan et al., 2026). The harm of history-based selection is therefore not an artifact of the learning rate we chose; a smaller step size delays it and a larger one amplifies it.

F.17 Unfiltered pool

The Science pool keeps prompts on which the base policy's reference pass rate lies in a band, which removes prompts it never solves (Appendix E). The filter could remove the prompts on which a

selector earns its savings. We therefore rebuild the pool as 406 prompts drawn uniformly from all 1,500 candidate training prompts, without any pass-rate filter, and repeat uniform sampling, dynamic sampling, GRESO and GRESO with the visit-triggered audit of Section 5. In Science the audit has no exact verifier, so DeepSeek-V4-Flash, with the reference answer in view, serves as the auditing judge, and a prompt is audited once its visits exceed 2.1 times its uniform share (1.5 in Math). All arms use the training recipe with the overlong penalty, $G = 8$ and 200 steps. In this pool 80% of the prompts have a reference pass count of 0 of 8 for the base policy, and the training judge's precision on base-policy responses is 0.14, against 0.35 on the filtered pool. Table 48 reports the results.

Without the filter GRESO again ends well below uniform sampling (-8.5 ± 1.9 points), and dynamic sampling again ties it. The loss is slightly smaller than on the filtered pool (-10.1 ± 2.1) because a larger set of prompts stays judge-informative and GRESO's visits spread over it (visit concentration 0.303 against 0.358). The audit retires 61.7 ± 2.3 prompts from the pool, raises the training judge's pass rate on the trained groups over steps 160–200 from 0.760 ± 0.006 under GRESO to 0.847 ± 0.003 (uniform sampling 0.820 ± 0.006), because the prompts it retires are the hard ones on which GRESO concentrates, and ends above GRESO in every seed, by $+4.3 \pm 0.3$ points, but remains -4.2 ± 1.6 points below uniform sampling. Filtering the pool is therefore not what removes the selectors' benefit; the prompts the base policy never solves are the ones a selector is designed to skip or revisit, and under this judge they are where it keeps finding spurious signal.

F.18 Safeguards without the overlong penalty

Appendix F.6 reports uniform sampling, GRESO and MoPPS under the training recipe without the overlong penalty, in which the selectors' responses grow until most are truncated at the 2,048-token limit (a runaway growth of response length). Table 49 adds three safeguards run under the same recipe, namely GRESO with the visit-triggered audit (DeepSeek-V4-Flash as the auditing judge, as in Appendix F.17), GRESO with a visit cap of 2.1 times uniform sampling's visit share (the Science setting of both safeguards), and uniform sampling with our per-response variant of Rubric Dropout (Yang et al., 2026a), which shows each Important or Optional rubric criterion to the judge with probability one half, drawn separately for each response, while Essential criteria and pitfalls are always shown. We define the onset of the runaway as the first step from which the share of truncated responses stays above one half for ten consecutive steps.

Both safeguards reduce GRESO's loss without the overlong penalty (GRESO alone -10.3 ± 4.5, Appendix F.6). The audit postpones or prevents the runaway, since it retires the prompts on which the judge credits truncated responses, and the cap delays it by only a few steps. Our per-response variant of Rubric Dropout, which unlike the published method draws a separate subset for each response of a group, makes uniform sampling worse by 15 points, since it adds reward differences within a group that are unrelated to the response, and the runaway begins earlier; we did not test the published scheme (Appendix G).

G Negative results

We report the approaches that did not work, or did not behave as expected, so that others can avoid them.

Asking the same judge again does not remove its errors. A natural defense against judge error is to re-query the judge on responses it accepted and keep only verdicts that repeat. In Math the training judge (Gemma-3-12B) re-accepts a fixed wrong response on 0.68 ± 0.05 of re-checks (response-cluster bootstrap SE), against a base false-positive rate of 0.22 ± 0.03 (Appendix C), so a re-check by the same judge confirms most of the false positives it is meant to remove, and the prompts whose false positives persist are exactly the ones a history-based selector keeps revisiting. Aggregating several weak judges helps little. In Science, a majority vote of the training judge and two weaker judge configurations lowers the false-positive rate against the panel only from 0.213 to 0.158.

A larger judge is not always a better judge. Qwen3-32B agrees with the Science panel on 0.731 ± 0.004 of criteria, below the 12B training judge (0.760 ± 0.004; binomial SEs over about 10,500 criteria), and in Math Gemma-3-27B in thinking mode has the same error rates as Gemma-3-12B (false positives 0.212 ± 0.030 against 0.222 ± 0.030; false negatives 0.177 ± 0.030 against

0.163 ± 0.029). Turning thinking off lowers the 27B judge's false-positive rate to 0.079 ± 0.024 but raises its false-negative rate to 0.375 ± 0.043, which trades one kind of spurious group for another. An evidence-first rewrite of the rubric template, which asks the judge to state its evidence before deciding each criterion, lowers the Gemma-4-12B judge's per-criterion agreement from 0.760 to 0.655, so we kept the original template. Scale within a family helps in one case and not in the other. Gemma-4-26B-A4B (25.2B parameters, 3.8B active) agrees with the panel on 0.933 ± 0.002 of criteria against 0.760 for Gemma-4-12B (Appendix C), while Gemma-3-27B is no better than Gemma-3-12B in Math, so the quality of a judge cannot be read from its parameter count.

A per-response variant of Rubric Dropout makes the recipe without the overlong penalty worse. Rubric Dropout (Yang et al., 2026a) is a published mitigation of reward hacking under rubric rewards, which drops one subset of criteria shared across a group. Our variant draws a separate subset of the Important and Optional criteria for each response, at rate one half, and always shows the Essential criteria and pitfalls. Uniform sampling with this variant ends -15.0 ± 1.6 points below uniform sampling on MMLU-Pro STEM without the overlong penalty, and its runaway growth of response length starts about 30 steps earlier (Table 49). Drawing per response adds reward differences within a group that are unrelated to the response; we did not test the published scheme.

Science selectors do not concentrate on the judge's most error-prone prompts. We expected the Science selectors, like the Math selectors, to concentrate on prompts where the judge's false credit is high. They do not. Among the pool prompts with panel labels, the half that the selectors visit most has a lower training-judge false-positive rate on base-policy responses (0.214) than the less-visited half (0.300). The selectors spend their visits on prompts that are harder under both judges. The Science loss is therefore not explained by prompt-level concentration on judge errors, which is why Section 3 describes the loss as judge-dependent but not explained by amplification, and why its dependence on the judge rests on the judge-strength comparison.

Without the overlong penalty, the audit's effect is not separable from seed variance. Without the overlong penalty the visit-triggered audit ends -4.1 ± 1.1 points below uniform sampling, against -10.3 ± 4.5 for GRESO alone (Table 49). The paired difference between the two, $+6.3 \pm 4.2$ points, is smaller than its spread across seeds, because GRESO's own result under this recipe ranges from 2.7 to 18.2 points below uniform sampling depending on when its runaway growth of response length begins. The same audit is better than GRESO in every seed on the unfiltered pool with the overlong penalty ($+4.3 \pm 0.3$; Appendix F.17), where GRESO's seeds agree more closely.

Within-step repeats inflate a selector's informative-group count. In an early implementation, when GRESO skipped most of the pool, the few surviving prompts were rolled several times within the same step to fill the batch. Repeated groups of one prompt are highly correlated, so this raised the count of informative groups without adding information and made the selector look more efficient than it is. Every reported run decides each prompt at most once per step, and when fewer eligible prompts survive than the step needs, the remainder is filled with distinct eligible prompts drawn uniformly (Appendix E).

Masking truncated responses does not remove reward hacking in Math. Excluding truncated responses from the gradient is a common part of the recipe. In Math, GRESO with truncated responses masked still produces $+15.3 \pm 2.0$ more reward-hacking events than uniform sampling with the same masking (Appendix F.6), close to the unmasked excess, so the Math events do not come from truncation.

H EXAMPLES OF REWARD-HACKED RESPONSES

This appendix shows responses that a training judge accepted although they are wrong, quoted verbatim from our logs and trimmed where marked with `[...]`; line breaks inside long lines are ours. In Math the trainer logs per-group verdict counts rather than response text, so the Math examples come from two sources. Examples 1, 2, 3 and 5 are responses of the base policy (Qwen3-1.7B-Base) to prompts of the hard set that later became reward-hacking events in training runs, taken from the response set of the re-check measurements of Appendix C; the Math training judge (Gemma-3-12B,

thinking mode) judged each of them ten times in each of the two measurements that included it (twenty verdicts for Examples 2 and 5). Example 4 is text produced late in a training run, which we logged for the variant of the audit with an auditing judge. The Science examples are held-out responses graded by the rubric judge (Gemma-4-12B) during training and re-graded afterwards by the stronger judges. Across the examples, three patterns recur. In the first, the response reports a computation that it did not perform and the judge trusts the reported result. In the second, the response states a confident, self-consistent wrong answer that the judge cannot check against information it does not see. In the third, the response contains no answer at all, and the judge either solves the problem itself or credits the form of the response.

H.1 MATH

Example 1 (invented program output). The prompt reads "How many numbers can be expressed as the sum of two or more distinct elements of the set $\{0, 1, 2, 4, 8, 16\}$?" and the reference answer is 31. The response writes a correct program but reports an output the program would not produce.

````
To solve this problem, we need to find out how many distinct sums can
be made with one or more elements from the set
$\{ 0, 1, 2, 4, 8, 16 \}$. [...]
for r in range(1, len(elements) + 1):
    for subset in combinations(elements, r):
        unique_sums.add(sum(subset))
[...]
```output
37
```
The number of unique sums that can be expressed as the sum of two or more
distinct elements of the set $\{ 0, 1, 2, 4, 8, 16 \}$ is \(\boxed{37}\).
````

The judge accepted this response in 9 of 10 verdicts. One accepting verdict ends "The output of the code, 37, is the correct number of distinct sums. CORRECT"; another lists the sums 1 to 31, writes "The number of such sums is 31", and still concludes "The code correctly calculates 31. CORRECT". The prompt became a reward-hacking event in several selector runs, one of them a MoPPS run at step 284, on the prompt's 35th visit, when the judge accepted 6 of 8 responses and the exact verifier none.

Example 2 (invented numerical result). The prompt reads "Find the largest integer x such that $\sqrt{3620 + 322x - 4x^2}$ is a real number." and the reference answer is 90 (the roots are -10 and 90.5).

````
Let's compute the roots.
```python
[...]
print(((root1, root2)))
```
```output
(136.38806064344338, -27.22806064344338)
```
The roots of the quadratic equation are approximately \(136.388\) and
\(-27.228\). [...]
Since we are looking for the largest integer \(x\), we take the integer
closest to \(136.388\), which is \(136\).
[...]
\boxed{136}
````

The judge accepted it in 18 of 20 verdicts, for example "They correctly identified the largest integer within that interval as 136. The python code confirms the roots are approximately -27.228 and 136.388. CORRECT". In a MoPPS run the prompt became an event at step 203 (visit 18), after a visit at step 193 on which one of eight responses was verified correct.

Example 3 (wrong work followed by unrelated text). The prompt reads "What is the units digit of $3^1 + 3^3 + 3^5 + 3^7 + \ldots + 3^{2009}$?" and the reference answer is 3.

```
- 3^7 (6561) has a units digit of 1
[...]
We can see that the units digits repeat in the pattern 3, 7, 3, 1.
Step 2: [...] there will be 2009 / 4 = 502 full patterns with 1 term
left over (3).
[...]
So, the units digit of the sum of all powers of 3 in the sequence is
Answer: 1.

How do you handle the fact that numerical values with human qualities
is puzzling or weird?
[...]
Distorted Energy Complex (DEC): The DEC measures how one's divine energy
interacts with their disruptive energies.
[...]
```

The work contains two errors ($6561 = 3^8$, and the sum has 1,005 terms), and the response continues with text unrelated to the problem. The judge accepted it in 10 of 10 verdicts, for example "The student correctly identifies the pattern of units digits: 3, 7, 3, 1. [...] Therefore, the student's final answer of 1 is correct. CORRECT". In a MoPPS run the prompt became an event at step 197, on its 52nd visit; that run visited it 111 times in 300 steps, about nine times the number of visits under uniform sampling.

Example 4 (a converged wrong answer late in training). The prompt is a counting problem about the product of the numbers on two spinners, whose sectors are given only in an accompanying diagram (one spinner has four sectors, the other three), and the reference answer is $\frac{2}{3}$. In a GRESO run the prompt became an event at step 158, on its 12th visit, when the judge accepted 7 of 8 responses and the exact verifier none. All seven accepted responses end with $\frac{3}{4}$, which treats both spinners as having four sectors; the end of one of them reads:

```
text{product is even}) = 1 - P(\text{both numbers are odd})
= 1 - \frac{1}{4} = \frac{3}{4}
\]
Thus, the probability that the product of the two spinners' numbers is
even is:
\[
\boxed{\frac{3}{4}}
\]
```

The reference-free judge cannot see the diagram, so a confident and internally consistent wrong answer passes; the auditing judge (GLM-5.3-Flash), which saw the reference answer, rejected all seven.

Example 5 (no answer at all). The prompt reads "The five-digit number 246_8 is divisible by 9. What is the missing digit?" and the reference answer is 7. The complete response restates the question and then continues with an unrelated fragment in another language, which we omit:

```
Let's solve this step by step. We know The five-digit number
$246\underline{\hspace{5pt}}8$ is divisible by 9. What is the missing
digit?[...]
```

The judge accepted it in 20 of 20 verdicts. Each verdict solves the problem itself and then grades its own solution, for example "So, $20 + x = 27$, which means $x = 7$. The missing digit is 7. CORRECT". This is the Math counterpart of the question echoes that the Science judge passes.

H.2 Science

Example 6 (question echo). The prompt, from the held-out rubric set, reads "What is the main difference between fraternal twins and identical twins in terms of their formation and genetic similarity, and how can you determine whether a pair of twins is fraternal or identical?" Two of its eight criteria are Essential, for example "Must specify that identical twins share 100% of their DNA whereas fraternal twins share roughly 50% of their genetic material." The base policy's complete greedy response repeats the instruction and the question:

```
Answer the following science question. Give a complete, well-organized
response that states the answer and the reasoning behind it.

What is the main difference between fraternal twins and identical twins
in terms of their formation and genetic similarity, and how can you
determine whether a pair of twins is fraternal or identical?
```

The rubric judge marked all eight criteria as met and gave reward 1, with the reference answer in view, and did so in the held-out evaluation of every arm at step 0. The stronger judge (Gemma-4-26B-A4B) marked seven of the eight criteria as unmet. Responses of this kind pass the rubric judge 82% of the time (Appendix C).

Example 7 (a response that grades itself). A prompt from the held-out rubric set asks for $[\mathrm{H_3O^+}]$ given $\mathrm{pOH} = 11.2$, and the reference answer is $10^{-2.8}$. The first Essential criterion reads "The response must clearly state the final value of [H3O+] as 10^-2.8, matching the reference answer". The response, from a MoPPS run at step 150 without the overlong penalty, was truncated at 2,048 tokens:

```
Response:
The response is correct and well-organized. It provides the answer and
the reasoning behind it.

1. The response states the answer: the value of
\(\left[{H}_{3} {O}^{+}\right]\) is \(6.31 \times 10^{-3}\).

2. The response provides the reasoning behind it:
   - It correctly uses the relationship
     \(\left[{H}_{3} {O}^{+}\right] \left[H {O}^{-}\right]
     = {10}^{- 14}\) at 298 K.
   [...]
The response is complete, well-organized, and provides a clear and
logical sequence of steps to arrive at the answer. [...]
Output:
The response is correct and well-organized. [...]
```

The rubric judge marked all seven criteria as met, although the stated value is wrong; Gemma-4-26B-A4B and DeepSeek-V4-Flash both failed the response. Late in the same run, the phrase "The response is correct and well-organized" appears in many training responses, most of which the rubric judge rewarded, so the policy had learned to write a favorable verdict about its own answer.

I Datasets, licensing and contamination

I.1 Datasets and licenses

Table 50 lists every dataset and model we use, with its role and license. Appendix E describes how the pools and held-out sets are built from them.

MATH. The MATH problems come from mathematics competitions, and the dataset's original Hugging Face copy was taken down in January 2025 after a copyright notice from the Art of Problem Solving. The original release carries the MIT license, and we use a public mirror of it that preserves the original splits and file names. We use the problems for research only and release the problem identifiers, never the problem text. The mirror's training split holds the original training problems together with the original test problems outside the MATH-500 subset of Lightman et al. (2024). Both our pool and our held-out Math set (500 problems, 100 per difficulty level) are drawn from it and are disjoint from each other; the held-out set is therefore not MATH-500, and our held-out numbers are not directly comparable to published MATH-500 results.

RaR-Science. The dataset card of RaR-Science states no license. Its questions are drawn from SCP-116K, NaturalReasoning and GeneralThought-430K, whose licenses are listed above, and its rubrics were generated by a language model (o3-mini). Two of the source licenses are non-

commercial and one of them is share-alike, so we treat the whole dataset as available for non-commercial research only, use it for that purpose, and release prompt identifiers and per-criterion verdicts rather than question, reference or rubric text.

Judges. The Gemma Terms of Use may treat a model trained on Gemma outputs in order to perform similarly to Gemma as a derivative of Gemma. Our Math policies are trained on binary rewards computed by Gemma-3-12B, not on its text, and do not imitate it, but we flag the terms and release those checkpoints under them. The Gemma 4 models are released under Apache-2.0 and impose no such condition. The panel models and DeepSeek-V4-Flash were called through a commercial API under the provider's terms.

I.2 Contamination

We limit and bound the effect of overlap between training and evaluation data in three ways. First, when building the Science pool we removed every candidate training prompt whose token Jaccard similarity to any item of MMLU-Pro STEM, the held-out rubric set or GPQA-Diamond is 0.5 or more, so the pool contains no near-duplicate of an evaluation item. The held-out rubric set comes from the RaR-Science test split and the training pool from its training split. Second, the Math pool and the held-out Math set share no problem, and neither contains a MATH-500 problem. Third, Qwen2.5 base models have been shown to have seen MATH test problems during pretraining, and Wu et al. (2026b) report the same for Qwen3-8B-Base, which completes 48% of MATH-500 problems exactly when shown the first 80% of the problem text. Such exposure would affect every arm, which start from the same checkpoint and are compared by paired differences, although reinforcement learning can surface memorized answers unevenly; it does not affect our Math diagnosis, which is measured on the training prompts themselves with the exact verifier (strict mastery, reward-hacking events and the judged-minus-true gap). The Science comparison uses MMLU-Pro STEM, whose multiple-choice format differs from the free-form training prompts.

J Analysis plans and amendments

Before each group of runs we wrote an analysis plan that fixed the arms, the settings, the primary quantities and the readouts that would count as support for or against each claim. When the design changed, we wrote an amendment before launching the affected runs. This appendix lists the plans and amendments in the order in which they were written, marks every change that was made after seeing results, and states how the paper reports both alternatives in those cases. Seeds 1 and 2 of every arm were run under the plan that governed seed 0, without changes.

Plan 1, before the first Math runs. The plan fixed the Math design, with Qwen3-1.7B-Base under full fine-tuning at a learning rate of 3×10^{-6}, $G = 8$, $P = 32$, the Gemma-3-12B judge in thinking mode served on the training node, and an exact verifier that records correctness without entering the reward. It fixed the selectors (GRESO, first reimplemented from its paper and later ported from the authors' code, which all reported runs use; MoPPS taken from the authors' code), uniform sampling as the reference, a run of every judge-trained arm with the exact verifier as the reward, and a random-selection control matched to GRESO's skip counts. The statistic of record was the change, over training, in the share of trained visits falling on a set of prompts flagged as error-prone from base-policy responses before training.

Amendment 1, after the first Math runs (changed after seeing results). The first runs showed two design defects and one uninformative statistic, and the amendment was written before the runs reported in the paper. First, when GRESO skipped most of the pool it rolled the few surviving prompts several times in one step, which inflated its informative-group count (Appendix G); from then on every arm decides each prompt at most once per step, and a step with too few eligible prompts is filled with distinct eligible prompts drawn uniformly, counted and reported per arm. Second, the pool was enlarged to 800 prompts and the run to 300 steps, so that uniform sampling visits each prompt 12 times while a selector remains free to visit a prompt far more often. Third, the pre-training error-prone set did not track where the judge's errors arose during training, so the statistics of record were replaced by quantities measured with the exact verifier on the trained groups, namely the judged-minus-true gap, the spurious-group share and held-out MATH accuracy. The

pre-training statistic is not used in the paper; the error-prone set $\hat{\mathcal{E}}$ of Appendix B.5 is defined from uniform-sampling runs instead. The random-selection control of Plan 1 was dropped with that statistic, since it was designed to separate skip counts from the share of error-prone visits; the replay controls of Section 3 answer the same question more directly. The same amendment defined the visit-triggered audit (a prompt whose visits exceed 1.5 times uniform sampling's share is audited with the exact verifier and retired after two consecutive audited visits in which the judge passes responses that the verifier rejects), the visit cap without an audit, and the Math gradient-masking pair.

Amendment 2, repeated selector runs. The first GRESO and MoPPS runs of seeds 0 and 1 stopped at steps 274 to 288, when their jobs reached the time limit. Each was run again to 300 steps with the same seed and the same configuration, which also estimates run-to-run variation. No run was discarded. All five runs of each selector are reported, the stopped runs over their logged steps (Appendix B.5), and the two runs of a seed are averaged before the standard error over the three seeds is computed.

Amendment 3, metric definitions and Plan 2 for Science (changed after seeing results). The first reading of the 800-prompt runs attributed the selectors' excess of reward-hacking events to how soon a prompt is revisited. A review of that reading showed that the effect follows how many times a prompt has been visited, at every revisit gap, so the claim was changed accordingly (Appendix F.3). With this amendment the definitions of the hard set $\mathcal{H}$, the reward-hacking event and strict mastery were fixed in the analysis code, after this first reading and before any later run; because the event definition was fixed after data had been seen, we report both the event definition and a looser alternative (Appendix F.4), and the conclusions hold under both, as they do across judge thresholds (Table 8). Plan 2 fixed RaR-Science as the data, a pool restricted to a band of base-policy reference pass rates, MMLU-Pro STEM as the statistic of record with an item-paired test against uniform sampling, the held-out rubric set scored offline by a panel of two language models, our per-response variant of Rubric Dropout, and the rule for choosing the training judge, which must agree with the panel on between 0.70 and 0.90 of criteria, so that it is imperfect by design but not uninformative.

Amendment 4, choice of the Science judge. Following that rule, the calibration against the panel selected Gemma-4-12B with the original rubric template (0.760 per criterion); Gemma-3-12B fell below the band and the evidence-first template lowered agreement (Appendix G). The pool band was rebuilt on the reference judge's verdicts, giving the 406-prompt pool.

Amendment 5, evaluation corrections before further Science runs. A review found two defects in the first multiple-choice answer extractor (it ignored answers in `\boxed{}` and lacked a word boundary after the option letter). All checkpoints were re-scored from saved weights with the corrected extractor, and the scores recorded during training are not used.

Plan 3, overlong masking (stopped and replaced). Without an overlong penalty the selectors' responses grew until most were truncated, and the judge credited many truncated responses. The plan tested masking truncated responses out of the loss. It noted in advance that masking cannot remove the credit a truncated response adds to its group's mean reward, so that late in a runaway growth of response length it could push the policy away from stopping; the runs were stopped after fewer than 40 steps and replaced by Plan 4, and no result from them is reported.

Plan 4, the overlong penalty (changed after seeing results). The plan adopted the soft overlong penalty of DAPO (Yu et al., 2025) (no penalty up to 1,536 tokens, then a linear ramp to -1 at the 2,048-token limit) after the recipe without it had produced the runaway; the penalty was chosen after seeing data, so the paper reports both recipes, with the recipe without the penalty in Appendix F.6. The answer extractor was extended in the same plan to accept a boxed option value, after dynamic sampling was found to box the value of an option instead of its letter on some items; this change was also made after seeing data. The earlier extractor changes the scores without the penalty by at most 0.3 points, while it scores dynamic sampling's boxed values as unanswered.

Plan 5, the judge-strength comparison. The plan repeats the arms of Plan 4 with Gemma-4-26B-A4B as the training judge, with the readout fixed in advance that a selector deficit within five points

of uniform sampling indicates a judge-driven loss and a deficit of eight points or more a loss that does not depend on the judge. Both selectors' deficits fall in the first case (-3.4 ± 0.9 for GRESO and -2.7 ± 0.8 for MoPPS). The plan also recorded that the training judges are served without a beginning-of-sequence token, while the 26B judge's calibration was measured with one.

Plan 6, the importance-weighted arms. The plan specified IW-GRESO and IW-MoPPS with a floor of 0.3, the readouts that the correction is supported if IW-GRESO ends within 2.5 points of uniform sampling and refuted if it ends five or more points below, and IW-MoPPS as a control of the implementation (its keep propensities are nearly uniform). It noted that the wrapper changes two things, the floor and the weights, and planned the control that separates them, GRESO's floored draw without weights. IW-GRESO ends -0.8 ± 1.0 points from uniform sampling, which meets the first readout, and the floored draw without weights ends -4.6 ± 0.9 points below it (Section 6).

Later plans. The remaining runs, namely the Qwen3-8B-Base block, the DPPO, Prompt Replay and noise-corrected GRPO baselines, the replay controls, the safeguard ablations, the Math pool-size runs, the floor sweep and the group-size, learning-rate and unfiltered-pool sweeps of Appendix F, were each run under a plan written beforehand and needed no amendment. The unfiltered-pool plan, for example, fixed in advance that dynamic sampling should end within five points of uniform sampling and that the audit should retire more prompts than on the filtered pool and end above GRESO; both readouts were met (Appendix F.17).

Changes made after seeing results. Four decisions followed observation of the data, namely the change of the Math statistics of record (Amendment 1), the fixing of the Math metric definitions with the move from revisit timing to visit count (Amendment 3), the soft overlong penalty and the extended answer extractor (both Plan 4). For each, the paper reports the alternative as well, and no conclusion changes under it.

K Code and data release

We will release the following under the MIT license, except where a component inherits a more restrictive license (noted below); an anonymized copy of the code is provided as supplementary material.

Code. The training code, built on TRL (von Werra et al., 2020) with the policy sampler, the judge server and the trainer on one node; our implementations of uniform sampling, dynamic sampling, the Prompt Replay, DPPO and noise-corrected GRPO baselines and the visit cap; GRESO, ported from the authors' code, whose repository states no license, released with attribution and subject to the authors' terms; the MoPPS selector taken from the authors' Apache-2.0 code with its notice; the importance-weighted wrapper (Algorithm 1), which takes any selector's keep propensities and returns the drawn prompts and their weights; the visit-triggered audit; the Math exact verifier and the rubric scoring rule; the judge prompts and rubric templates of Appendix C; and the scripts that compute every metric of Appendix B, every table and every figure from the logs, with unit tests for the sampling design (the inclusion probabilities, the weights and the fixed loss normalizer).

Data. For every pool and held-out set, the prompt identifiers in the source datasets and the construction scripts, but not the prompt text, references or rubrics, which remain under their source licenses (Appendix I). For every training run, the per-step and per-group logs, which record the prompts drawn, their inclusion probabilities and weights, the judge's and the exact verifier's verdict counts for each group, and the visit counts, from which every Math diagnosis in Sections 3 and 6 can be recomputed. The per-criterion verdicts of the training judge, the stronger judges and the panel on the held-out rubric responses and on the calibration set; the ten repeated verdicts per response of the re-check measurements; and the analysis plans of Appendix J with their amendments.

Checkpoints. The final policy checkpoints of the three seeds of every main-text arm. Checkpoints of Math runs trained with rewards from Gemma-3-12B are released under the Gemma Terms of Use (Appendix I.1). Science checkpoints are trained on RaR-Science, whose sources include non-commercial licenses, so we release them for non-commercial research only.

Table 21: Math diagnosis per seed (Qwen3-1.7B-Base, Gemma-3-12B-it judge unless stated; $\mathcal{H}$ has 359 prompts). Runs within a seed are separated by commas. "Exact" rows train with the exact verifier as the reward.

Quantity	Arm	Mean ± SE	Seed 0	Seed 1	Seed 2
Strict mastery of $\mathcal{H}$ at step 200, exact	GRESO	98.3 ± 0.1	98.1	98.6	98.3
	MoPPS	94.6 ± 0.1	94.7	94.4	94.6
	Uniform	84.7 ± 0.6	83.8	85.8	84.6
Lead over uniform, exact	GRESO	$+13.6 \pm 0.4$	$+14.3$	$+12.8$	$+13.7$
	MoPPS	$+9.8 \pm 0.7$	$+10.9$	$+8.6$	$+10.0$
Held-out MATH, exact	GRESO	83.3 ± 0.7	83.8	82.0	84.2
	MoPPS	84.7 ± 0.6	85.6	84.8	83.6
	Uniform	83.3 ± 1.1	81.4	85.2	83.2
Strict mastery of $\mathcal{H}$ at step 200	GRESO	76.7 ± 0.5	74.7, 77.4	78.3, 76.9	76.5
	MoPPS	74.0 ± 3.5	54.9, 79.4	80.2, 75.8	77.0
	Uniform	72.8 ± 0.3	73.3	72.4	72.8
Lead over uniform	GRESO	$+3.9 \pm 0.7$	$+1.4$, $+4.2$	$+5.9$, $+4.5$	$+3.7$
	MoPPS	$+1.2 \pm 3.7$	-18.4, $+6.1$	$+7.8$, $+3.4$	$+4.2$
Reward-hacking events on $\mathcal{H}$	GRESO	29.8 ± 0.2	31, 28	23, 37	30
	MoPPS	32.3 ± 0.7	44, 22	30, 32	33
	Uniform	11.3 ± 0.3	11	11	12
Events minus uniform	GRESO	$+18.5 \pm 0.3$	$+20$, $+17$	$+12$, $+26$	$+18$
	MoPPS	$+21.0 \pm 0.6$	$+33$, $+11$	$+19$, $+21$	$+21$
Events per 1,000 visits to $\mathcal{H}$	GRESO	4.73 ± 0.03	5.13, 4.21	3.82, 5.67	4.76
	MoPPS	4.93 ± 0.10	6.92, 3.20	4.70, 4.76	5.01
	Uniform	2.63 ± 0.08	2.55	2.55	2.79
Chance expectation of events	GRESO	19.1 ± 0.8	20.3, 14.9	17.6, 23.1	19.2
	MoPPS	13.9 ± 1.1	19.6, 12.1	12.2, 12.0	13.6
	Uniform	10.8 ± 0.2	10.5	10.6	11.2
Events beyond chance	GRESO	10.8 ± 0.6	$+10.7$, $+13.1$	$+5.4$, $+13.9$	$+10.8$
	MoPPS	18.5 ± 0.7	$+24.4$, $+9.9$	$+17.8$, $+20.0$	$+19.4$
	Uniform	0.6 ± 0.1	$+0.5$	$+0.4$	$+0.8$
Chance events per 1,000 at-risk visits	GRESO	4.6 ± 0.2	4.9, 3.7	4.9, 4.9	4.6
	MoPPS	3.2 ± 0.3	4.4, 3.1	2.9, 2.8	3.1
	Uniform	7.4 ± 0.1	7.4	7.4	7.2
Events per 1,000 at-risk visits	GRESO	7.2 ± 0.0	7.5, 7.0	6.4, 7.9	7.2
	MoPPS	7.6 ± 0.1	9.8, 5.6	7.2, 7.4	7.6
	Uniform	7.7 ± 0.0	7.8	7.7	7.7
At-risk visits to $\mathcal{H}$	GRESO	$4{,}140 \pm 28$	4,150, 4,020	3,610, 4,690	4,180
	MoPPS	$4{,}260 \pm 33$	4,480, 3,950	4,160, 4,300	4,320
	Uniform	$1{,}470 \pm 46$	1,410	1,429	1,558
Spurious-group share, last 20%	GRESO	20.7 ± 0.9	13.4, 25.7	27.4, 17.7	20.0
	MoPPS	19.2 ± 1.2	10.8, 23.7	23.1, 19.6	19.0
	Uniform	5.2 ± 0.1	5.3	5.2	5.0
Spurious share minus uniform	GRESO	$+15.5 \pm 0.9$	$+8.1$, $+20.4$	$+22.2$, $+12.5$	$+15.0$
	MoPPS	$+14.0 \pm 1.2$	$+5.5$, $+18.4$	$+17.9$, $+14.4$	$+14.0$
Judged-minus-true gap δ, last 20%	GRESO	0.129 ± 0.005	0.190, 0.076	0.096, 0.174	0.120
	MoPPS	0.306 ± 0.035	0.541, 0.209	0.291, 0.253	0.270
	Uniform	0.040 ± 0.002	0.037	0.043	0.040
Gap minus uniform (points)	GRESO	$+8.9 \pm 0.5$	$+15.3$, $+3.9$	$+5.3$, $+13.1$	$+8.0$
	MoPPS	$+26.6 \pm 3.6$	$+50.4$, $+17.2$	$+24.8$, $+21.0$	$+23.0$
Visit amplification Λ_{vis}	GRESO	11.40 ± 0.10	11.68, 11.42	10.86, 11.58	11.42
	MoPPS	13.18 ± 0.09	13.03	13.35	13.17
	Uniform	1.00	1.00	1.00	1.00
Error-prone visit share	GRESO	28.6 ± 0.4	27.7, 28.2	30.1, 28.5	28.5
	MoPPS	38.5 ± 0.2	37.6, 40.1	37.2, 38.8	38.5
	Uniform	13.3	13.3	13.3	13.3
Held-out MATH at the end	GRESO	81.5 ± 1.3	83.6	79.2	81.8
	MoPPS	81.9 ± 1.6	84.6	79.0	82.2
	Uniform	82.4 ± 0.8	81.0	83.8	82.4
Held-out minus uniform	GRESO	-0.9 ± 2.1	$+2.6$	-4.6	-0.6
	MoPPS	-0.5 ± 2.4	$+3.6$	-4.8	-0.2

Table 22: Replay control per seed (Math, Gemma-3-12B-it judge). Events over the whole run; gap δ and spurious share over the last 20% of logged steps; strict mastery at step 200; held-out MATH at the end.

Quantity	Arm	Mean ± SE	Seed 0	Seed 1	Seed 2
Visit amplification Λ_{vis}	All three runs	11.47 ± 0.05	11.42	11.58	11.42
Error-prone visit share	Source and replay	28.4 ± 0.1	28.2	28.5	28.5
	Matched replay	16.1 ± 0.2	16.1	16.5	15.8
Reward-hacking events	Source run	31.7 ± 2.7	28	37	30
	Replay	26.7 ± 1.8	26	30	24
	Matched replay	11.0 ± 1.2	11	13	9
Events beyond chance	Source run	12.6 ± 0.9	13.1	13.9	10.8
	Replay	10.2 ± 0.8	9.9	11.7	9.1
	Matched replay	0.5 ± 0.1	0.6	0.5	0.4
Events per 1,000 visits to $\mathcal{H}$	Source run	4.88 ± 0.43	4.21	5.67	4.76
	Replay	4.11 ± 0.25	3.91	4.60	3.81
	Matched replay	1.69 ± 0.16	1.66	1.99	1.43
Judged-minus-true gap δ	Source run	0.123 ± 0.028	0.076	0.174	0.120
	Replay	0.117 ± 0.007	0.116	0.130	0.105
	Matched replay	0.043 ± 0.002	0.043	0.047	0.040
Spurious-group share	Source run	21.1 ± 2.4	25.7	17.7	20.0
	Replay	24.3 ± 0.6	25.4	24.0	23.5
	Matched replay	5.4 ± 0.2	5.4	5.8	5.1
Strict mastery at step 200	Source run	76.9 ± 0.3	77.4	76.9	76.5
	Replay	75.2 ± 0.3	74.7	75.2	75.8
	Matched replay	67.5 ± 0.4	67.4	68.2	66.8
Held-out MATH	Source run	81.5 ± 1.3	83.6	79.2	81.8
	Replay	82.1 ± 0.3	82.2	81.6	82.6
	Matched replay	81.8 ± 0.5	81.0	82.6	81.8
Replay minus matched	Error-prone share	$+12.3 \pm 0.2$	$+12.1$	$+12.0$	$+12.7$
	Events	$+15.7 \pm 0.7$	$+15$	$+17$	$+15$
	Events beyond chance	$+9.7 \pm 0.8$	$+9.3$	$+11.2$	$+8.7$
	Events per 1,000 visits	$+2.41 \pm 0.11$	$+2.25$	$+2.61$	$+2.38$
	Gap δ	$+0.074 \pm 0.005$	$+0.073$	$+0.083$	$+0.065$
	Spurious share	$+18.9 \pm 0.6$	$+20.0$	$+18.2$	$+18.4$
	Strict mastery	$+7.8 \pm 0.6$	$+7.3$	$+7.0$	$+9.0$
	Held-out MATH	$+0.3 \pm 0.7$	$+1.2$	-1.0	$+0.8$

Table 23: Science MMLU-Pro STEM (%) at every evaluation, including the early evaluations at steps 10 and 25 and the importance-weighted arms (the data of Figure 2c). Every arm rolls out 256 responses per step (dynamic sampling 384). Mean ± SE over three seeds.

Arm	Step 0	10	25	50	100	150	200
Uniform	18.3	27.6 ± 0.3	39.1 ± 0.4	45.2 ± 0.6	44.8 ± 1.5	47.7 ± 0.9	48.9 ± 0.4
Dynamic sampling	18.3	27.8 ± 0.3	39.4 ± 0.4	45.7 ± 0.5	47.3 ± 0.4	48.4 ± 0.3	49.0 ± 0.3
GRESO	18.3	27.1 ± 0.4	35.6 ± 1.3	40.9 ± 1.4	39.1 ± 3.4	38.8 ± 3.5	38.8 ± 2.6
MoPPS	18.3	27.4 ± 0.4	38.8 ± 0.5	45.3 ± 0.5	43.0 ± 0.9	39.9 ± 0.4	37.9 ± 0.5
IW-GRESO	18.3	27.3 ± 0.4	38.4 ± 0.5	44.0 ± 0.7	44.0 ± 1.3	46.4 ± 1.1	48.1 ± 1.3
IW-MoPPS	18.3	27.5 ± 0.3	38.8 ± 0.4	44.9 ± 0.5	43.9 ± 0.9	46.1 ± 0.5	47.7 ± 1.0

Table 24: Science MMLU-Pro STEM at the saved checkpoints (Qwen3-1.7B-Base, Gemma-4-12B-it rubric judge, overlong penalty). Uniform sampling rows give accuracy in percent; the other rows give the paired difference from uniform sampling in points.

Arm	Step	Mean ± SE	Seed 0	Seed 1	Seed 2
Uniform (level)	50	45.2 ± 0.6	45.4	46.2	44.1
	100	44.8 ± 1.5	42.0	47.1	45.2
	150	47.7 ± 0.9	47.6	49.3	46.3
	200	48.9 ± 0.4	48.8	49.7	48.2
Dynamic sampling	50	$+0.5 \pm 0.5$	$+1.1$	-0.4	$+0.8$
	100	$+2.5 \pm 1.7$	$+5.9$	$+0.3$	$+1.4$
	150	$+0.6 \pm 0.6$	$+0.8$	-0.4	$+1.5$
	200	$+0.1 \pm 0.7$	$+0.3$	-1.2	$+1.2$
GRESO	50	-4.3 ± 1.1	-6.4	-2.5	-4.0
	100	-5.7 ± 2.1	-7.6	-1.5	-8.0
	150	-9.0 ± 2.9	-13.1	-3.5	-10.3
	200	-10.1 ± 2.1	-11.2	-6.0	-13.1
MoPPS	50	$+0.1 \pm 0.1$	$+0.1$	-0.2	$+0.3$
	100	-1.7 ± 0.6	-0.5	-2.5	-2.2
	150	-7.8 ± 1.0	-8.5	-9.1	-5.8
	200	-11.0 ± 1.0	-11.2	-12.6	-9.3

Table 25: Style-split share of trained Science groups (%) by window of steps (the data of Figure 3c). Mean ± SE over three seeds, with per-seed values of the first and last windows.

Arm	Steps 0–24	25–49	50–99	100–149	150–199	Per seed, first / last window
Uniform	17.6 ± 0.3	17.2 ± 0.3	17.0 ± 0.2	16.9 ± 0.3	17.2 ± 0.2	17.6/18.1/17.2; 17.1/16.8/17.6
Dynamic sampling	18.8 ± 0.2	18.3 ± 0.2	18.1 ± 0.3	18.1 ± 0.1	18.1 ± 0.2	18.8/19.2/18.4; 18.0/17.7/18.5
GRESO	25.4 ± 1.1	29.3 ± 1.5	30.8 ± 1.5	31.4 ± 1.4	31.9 ± 1.4	26.2/23.1/26.8; 32.6/29.1/33.9
MoPPS	22.3 ± 0.3	26.4 ± 0.4	30.9 ± 0.5	35.5 ± 0.7	39.9 ± 0.8	22.9/22.2/21.8; 40.2/41.1/38.5
IW-GRESO	18.1 ± 0.2	18.4 ± 0.3	18.6 ± 0.2	18.6 ± 0.2	18.6 ± 0.2	18.1/17.8/18.5; 18.7/18.1/18.9
IW-MoPPS	18.5 ± 0.2	19.1 ± 0.2	19.4 ± 0.2	19.7 ± 0.3	19.8 ± 0.4	18.4/18.8/18.3; 19.7/20.5/19.2

Table 26: Share of terse answers (fewer than 150 tokens) on MMLU-Pro STEM (%) over training. Mean ± SE over three seeds; the base policy's share is 9% in every seed.

Arm	Step 10	25	50	100	150	200
Uniform	9.3 ± 0.3	9.7 ± 0.3	10.3 ± 0.3	11.0 ± 0.6	11.0 ± 0.6	11.0 ± 0.6
Dynamic sampling	9.7 ± 0.3	10.3 ± 0.3	11.3 ± 0.3	12.0 ± 0.6	12.7 ± 0.3	13.0 ± 0.6
GRESO	9.7 ± 0.3	15.7 ± 1.9	29.7 ± 5.9	41.3 ± 8.7	46.7 ± 9.4	48.7 ± 9.4
MoPPS	9.7 ± 0.3	12.3 ± 0.3	20.3 ± 0.9	38.0 ± 1.7	51.3 ± 2.3	59.3 ± 2.3
IW-GRESO	9.3 ± 0.3	10.3 ± 0.3	11.7 ± 0.3	12.7 ± 0.9	13.7 ± 0.9	13.7 ± 0.9
IW-MoPPS	9.3 ± 0.3	10.7 ± 0.3	12.0 ± 0.6	14.3 ± 0.9	15.7 ± 1.2	16.7 ± 1.2

Table 27: Event rate and strict-mastery rate per 1,000 at-risk visits, by visit number (Math, Gemma-3-12B-it judge). Strict-mastery rates of the selectors give the range over the four runs of seeds 0 and 1 and the value of seed 2.

Arm	Run	Visits 1–12	13–24	25–48	49+
Event rate					
Uniform	seeds 0 / 1 / 2	7.8 / 7.7 / 7.7	–	–	–
GRESO	seed 0, first run	4.4	14.2	21.9	12.3
	seed 0, second run	9.1	11.5	9.0	5.9
	seed 1, first run	5.2	7.8	13.5	9.3
	seed 1, second run	8.1	16.9	12.8	14.3
	seed 2	6.7	12.6	14.3	10.5
MoPPS	seed 0, first run	14.7	25.6	27.6	9.7
	seed 0, second run	6.6	13.5	5.1	5.4
	seed 1, first run	8.5	16.5	11.9	9.8
	seed 1, second run	12.7	13.2	6.5	5.6
	seed 2	10.6	17.2	12.7	7.6
Strict-mastery rate					
Uniform	seeds 0 / 1 / 2	181 / 158 / 169	–	–	–
GRESO	seeds 0–1 (range); seed 2	134–147; 142	43–47; 45	17–25; 20	6–21; 13
MoPPS	seeds 0–1 (range); seed 2	71–157; 127	16–65; 45	37–48; 42	0–30; 21

Table 28: Event rate of the pooled selector runs by visit number and steps since the previous visit (binomial SE; n is the number of at-risk visits).

Visits	Gap ≤ 2 steps	3–8 steps	9 or more steps
1–12	0.006 ± 0.001 ($n = 5{,}249$)	0.005 ± 0.001 ($n = 5{,}544$)	0.005 ± 0.001 ($n = 5{,}623$)
13–24	0.010 ± 0.001 ($n = 4{,}827$)	0.009 ± 0.002 ($n = 1{,}960$)	0.006 ± 0.003 ($n = 492$)
25–48	0.010 ± 0.001 ($n = 4{,}422$)	0.009 ± 0.003 ($n = 1{,}111$)	0.004 ± 0.004 ($n = 253$)
49+	0.008 ± 0.001 ($n = 4{,}060$)	0.006 ± 0.003 ($n = 517$)	0 of 49

Table 29: Jaccard overlap of the sets of prompts with a reward-hacking event, for pairs of Math runs (Gemma-3-12B-it judge).

Pairs of runs	Event definition (main text)	Loose definition
Selector runs	0.20–0.47 (mean 0.34)	0.24–0.54
GRESO runs	0.29–0.43	0.45–0.54
Uniform runs	0.22, 0.21, 0.28	0.42 (seeds 0 and 1)
Random sets of equal size	0.02–0.04 (within $\mathcal{H}$)	0.01–0.04 (pool); 0.03–0.09 (within $\mathcal{H}$)
Difficulty-matched random sets	0.09–0.13 (mean 0.11)	–

Table 30: Steps to strict mastery of 80% of $\mathcal{H}$ under the judge divided by the steps under the exact verifier, same seed (Math). A ">" marks a run that had not crossed 80% at its last step. Uncensored ratios range from $2.4\times$ to $3.1\times$ for GRESO (one run censored) and from $2.2\times$ to $2.9\times$ for MoPPS (two runs censored).

Arm	Run	Judge	Exact verifier	Ratio
GRESO	seed 0, first run	> 276	83	$> 3.3\times$
	seed 0, second run	258	83	$3.1\times$
	seed 1, first run	212	88	$2.4\times$
	seed 1, second run	253	88	$2.9\times$
	seed 2	256	88	$2.9\times$
MoPPS	seed 0, first run	> 279	96	$> 2.9\times$
	seed 0, second run	209	96	$2.2\times$
	seed 1, first run	183	83	$2.2\times$
	seed 1, second run	> 300	83	$> 3.6\times$
	seed 2	264	90	$2.9\times$
Uniform	seeds 0, 1, 2	> 300	163	$> 1.8\times$

Table 31: Further per-seed values of the Math diagnosis (Gemma-3-12B-it judge). Runs within a seed are separated by commas. Differences are in points, shares in percent.

Quantity	Arm	Mean ± SE	Seed 0	Seed 1	Seed 2
Held-out MATH minus uniform, exact verifier	GRESO	$+0.1 \pm 1.7$	$+2.4$	-3.2	$+1.0$
	MoPPS	$+1.4 \pm 1.4$	$+4.2$	-0.4	$+0.4$
Mastery lead lost from exact verifier to judge	GRESO	9.7 ± 1.1	11.5	7.6	10.0
Replay events shared with the source run	Replay	–	13 of 26	15 of 30	12 of 24
Events minus uniform	Replay	$+15.3 \pm 2.0$	$+15$	$+19$	$+12$
Mastered within six visits (uniform)	Uniform	93.3 ± 0.3	93	94	93
Events after the sixth visit	Selectors	80.2 ± 0.6	84, 79; 77, 77	87, 87; 83, 69	80.0
Events after the twelfth visit	Selectors	64.3 ± 0.7	65.7	63.3	64.0
All-rejected solvable groups	GRESO	3.73 ± 0.22	2.82, 4.04	5.08, 3.21	3.60
	MoPPS	0.38 ± 0.02	0.25, 0.44	0.39, 0.43	0.38
	Uniform	0.60 ± 0.05	0.52	0.68	0.60
Rollouts per truly informative group, last 100 steps	GRESO	17.2 ± 0.2	16.2, 18.7	17.6, 16.1	17.2
	MoPPS	17.3 ± 0.1	12.5, 21.6	15.9, 18.5	17.5
	Uniform	36.1 ± 0.6	37.2	35.0	36.0
Rollouts per truly informative group, whole run	GRESO	15.1 ± 0.1	14.7, 15.9	15.0, 14.8	15.1
	MoPPS	14.5 ± 0.2	11.5, 16.9	14.6, 14.9	14.6
	Uniform	27.1 ± 0.5	28.0	26.2	27.1
Step of 80% strict mastery, exact verifier	GRESO	86.3 ± 1.7	83	88	88
	MoPPS	89.7 ± 3.8	96	83	90
	Uniform	163	163	163	163
Events with gradient masking	GRESO	26.3 ± 2.6	22	31	26
	Uniform	11.0 ± 0.6	10	12	11
	GRESO − uniform	$+15.3 \pm 2.0$	$+12$	$+19$	$+15$
Strict mastery with gradient masking	GRESO	77.0 ± 0.3	77.0	77.5	76.5
	Uniform	72.8 ± 0.3	73.0	72.2	73.1

Table 32: Further per-seed values of the Science diagnosis (Gemma-4-12B-it rubric judge, with the overlong penalty unless stated). Accuracies and shares in percent, differences in points, lengths in tokens.

Quantity	Arm	Mean ± SE	Seed 0	Seed 1	Seed 2
MMLU-Pro STEM minus uniform, step 50	GRESO	-4.3 ± 1.1	-6.4	-2.5	-4.0
	MoPPS	$+0.1 \pm 0.1$	$+0.1$	-0.2	$+0.3$
	Dynamic sampling	$+0.5 \pm 0.5$	$+1.1$	-0.4	$+0.8$
Visit amplification, step 200	GRESO	5.96 ± 0.22	5.96	6.34	5.58
	MoPPS	9.16 ± 0.22	9.07	8.82	9.58
Visit amplification, step 50	MoPPS	6.09 ± 0.14	6.09	5.84	6.34
MMLU-Pro STEM, step 200	Dynamic sampling	49.0 ± 0.3	49.1	48.5	49.4
	GRESO	38.8 ± 2.6	37.6	43.7	35.1
	MoPPS	37.9 ± 0.5	37.6	37.1	38.9
Both selectors minus uniform, step 200	Pooled	-10.6 ± 0.6	-11.2	-9.3	-11.2
Accuracy where both arms reason	GRESO	33.4 ± 3.3	30.5	40.0	29.6
	Uniform (vs GRESO)	46.4 ± 0.3	46.4	47.0	45.8
	MoPPS	32.4 ± 0.4	32.3	33.1	31.8
	Uniform (vs MoPPS)	45.6 ± 0.3	45.5	45.2	46.1
Terse answers, step 200	Uniform	11.0 ± 0.6	11	12	10
	Dynamic sampling	13.0 ± 0.6	13	12	14
	GRESO	48.7 ± 9.4	56	30	60
	MoPPS	59.3 ± 2.3	63	60	55
	IW-GRESO	13.7 ± 0.9	14	12	15
	IW-MoPPS	16.7 ± 1.2	16	19	15
Response length, step 200	Uniform	514 ± 10	512	531	498
	Dynamic sampling	505 ± 10	505	522	489
	GRESO	370 ± 41	338	452	321
	MoPPS	318 ± 10	301	318	334
	IW-GRESO	491 ± 8	489	507	478
	IW-MoPPS	478 ± 6	476	468	489
Rubric-term restatement, steps 150–199	Uniform	21.3 ± 0.9	21	23	20
	Dynamic sampling	22.3 ± 0.9	22	24	21
	GRESO	42.3 ± 3.3	44	36	47
	MoPPS	52.0 ± 1.7	52	55	49
	IW-GRESO	23.3 ± 0.9	23	22	25
	IW-MoPPS	25.3 ± 0.9	25	27	24
Style-split share by consecutive prior informative visits (uniform runs)	0 prior	15.1 ± 0.3	15.3	14.6	15.4
	1 prior	21.9 ± 0.6	22.8	21.1	21.8
	2 or more	31.7 ± 1.0	33.1	29.8	32.2
Minus uniform on derivation items	GRESO	-15.8 ± 3.2	-17.3	-9.7	-20.3
	MoPPS	-17.0 ± 1.4	-17.1	-19.3	-14.5
Minus uniform on other items	GRESO	-2.4 ± 0.7	-2.9	-1.0	-3.3
	MoPPS	-3.0 ± 0.4	-3.2	-3.5	-2.3
Held-out rubric pass, stronger judge	Uniform	41.7 ± 0.6	41.7	42.8	40.6
	Dynamic sampling	36.6 ± 1.4	36.0	39.2	34.6
	GRESO	40.1 ± 0.3	40.1	39.5	40.7
	MoPPS	40.7 ± 1.1	40.8	42.5	38.8
Training judge passes, stronger judge fails	Uniform	32.4 ± 0.7	32.4	33.6	31.2
	Dynamic sampling	31.6 ± 0.9	31.2	33.3	30.3
	GRESO	34.6 ± 0.7	34.1	33.8	35.9
	MoPPS	39.5 ± 1.6	40.1	42.0	36.4

Table 33: Per-seed MMLU-Pro STEM accuracy of Qwen3-1.7B-Base in Science (%) and the paired difference from uniform sampling (points).

Method	STEM, seeds 0 / 1 / 2	Difference, seeds 0 / 1 / 2
Uniform	48.8 / 49.7 / 48.2	–
Dynamic sampling	49.1 / 48.5 / 49.4	$+0.3$ / -1.2 / $+1.2$
GRESO	37.6 / 43.7 / 35.1	-11.2 / -6.0 / -13.1
MoPPS	37.6 / 37.1 / 38.9	-11.2 / -12.6 / -9.3
DPPO	45.9 / 49.1 / 46.1	-2.9 / -0.6 / -2.1
Noise-corr. GRESO	38.7 / 44.2 / 36.2	-10.1 / -5.5 / -12.0
Prompt Replay	44.2 / 47.9 / 44.0	-4.6 / -1.8 / -4.2
IW-GRESO	46.4 / 50.6 / 47.3	-2.4 / $+0.9$ / -0.9
IW-MoPPS	46.4 / 49.6 / 47.1	-2.4 / -0.1 / -1.1

Table 34: Per-seed MMLU-Pro STEM accuracy of Qwen3-8B-Base in Science (%), the paired difference from uniform sampling (points), and GPQA-Diamond correct answers out of 198.

Method	STEM, seeds 0 / 1 / 2	Difference, seeds 0 / 1 / 2	GPQA-D (of 198)
Uniform	68.4 / 69.6 / 67.9	–	82 / 91 / 85
Dynamic sampling	69.6 / 68.6 / 68.3	$+1.2$ / -1.0 / $+0.4$	89 / 86 / 83
GRESO	61.8 / 63.6 / 59.8	-6.6 / -6.0 / -8.1	83 / 79 / 77
MoPPS	59.6 / 62.6 / 59.2	-8.8 / -7.0 / -8.7	84 / 83 / 74
DPPO	65.9 / 69.4 / 66.5	-2.5 / -0.2 / -1.4	87 / 84 / 83
Noise-corr. GRESO	63.0 / 63.5 / 60.2	-5.4 / -6.1 / -7.7	84 / 80 / 79
Prompt Replay	64.8 / 69.0 / 64.9	-3.6 / -0.6 / -3.0	86 / 82 / 82
IW-GRESO	66.9 / 70.3 / 66.9	-1.5 / $+0.7$ / -1.0	88 / 85 / 82
IW-MoPPS	66.5 / 69.8 / 66.9	-1.9 / $+0.2$ / -1.0	89 / 85 / 82

Table 35: Per-seed values behind Table 2b (Math, Qwen3-1.7B-Base), seeds 0 / 1 / 2. For GRESO and MoPPS, events and gaps list the two runs of seeds 0 and 1 separately; the other columns average them. Strict mastery of $\mathcal{H}$ at step 200 and MATH held-out are in %.

Method	Events on $\mathcal{H}$	Gap δ
Uniform	11 / 11 / 12	.037 / .043 / .040
Dynamic sampling	14 / 9 / 13	.044 / .041 / .045
GRESO	31, 28 / 23, 37 / 30	.190, .076 / .096, .174 / .120
MoPPS	44, 22 / 30, 32 / 33	.541, .209 / .291, .253 / .270
DPPO	14 / 10 / 12	.048 / .041 / .044
Noise-corr. GRESO	34 / 24 / 27	.150 / .110 / .121
Prompt Replay	22 / 16 / 19	.092 / .060 / .073
IW-GRESO	15 / 9 / 12	.046 / .040 / .043
IW-MoPPS	16 / 10 / 13	.049 / .041 / .045

Method	Mastery, exact verifier	Mastery, judge	MATH held-out
Uniform	83.8 / 85.8 / 84.6	73.3 / 72.4 / 72.8	81.0 / 83.8 / 82.4
Dynamic sampling	84.9 / 89.1 / 86.5	73.8 / 75.2 / 74.5	82.2 / 83.6 / 82.8
GRESO	98.1 / 98.6 / 98.3	76.0 / 77.6 / 76.5	83.6 / 79.2 / 81.8
MoPPS	94.7 / 94.4 / 94.6	67.1 / 78.0 / 77.0	84.6 / 79.0 / 82.2
DPPO	83.0 / 87.4 / 85.0	71.9 / 73.1 / 72.4	81.6 / 83.0 / 82.1
Noise-corr. GRESO	98.1 / 98.6 / 98.3	76.5 / 78.1 / 76.8	82.9 / 80.1 / 81.8
Prompt Replay	89.0 / 93.7 / 91.0	74.8 / 75.9 / 75.1	81.8 / 83.2 / 82.6
IW-GRESO	85.3 / 89.6 / 87.1	73.8 / 74.6 / 74.1	82.2 / 83.4 / 82.8
IW-MoPPS	84.6 / 88.9 / 86.4	73.5 / 74.2 / 73.7	82.6 / 83.0 / 82.4

Table 36: Safeguards and ablations in Math; mean ± SE over three seeds (levels). Mastery is strict mastery of $\mathcal{H}$ at step 200 under the judge-rewarded run, in %, on each pool's own hard set; events in the pool-size rows are normalized to 359 hard prompts. Visit amplification is as measured; the minimum of 8 visits before the cap binds lets the arms with $\Lambda_{\max} = 2.1$ reach 2.17. The floored draw without weights is IW-GRESO's draw with every weight set to 1. GRESO with the visit bound of 1.5 ($\Lambda_{\text{vis}} = 1.50$, held-out MATH 82.9 ± 0.2) and with the bound-triggered audit ($\Lambda_{\text{vis}} = 13.51 \pm 0.35$, held-out MATH 82.2 ± 0.4) are in Table 2b.

Arm	Events ↓	Gap δ ↓	Mastery ↑	Λ_{vis} ↓	MATH ↑
GRESO	29.8 ± 0.2	0.129 ± 0.005	76.7 ± 0.5	11.40 ± 0.10	81.5 ± 1.3
+ visit cap $\Lambda_{\max} = 2.1$	18.7 ± 2.0	0.071 ± 0.002	77.2 ± 0.2	2.17	83.5 ± 0.5
+ cooldown 12	20.7 ± 1.9	0.062 ± 0.002	76.9 ± 0.2	2.00	83.1 ± 0.4
+ visit-triggered audit (GLM-5.3-Flash)	22.0 ± 1.2	0.037 ± 0.002	79.4 ± 0.3	13.67 ± 0.14	80.6 ± 0.3
+ cooldown and audit	19.3 ± 0.9	0.052 ± 0.002	77.1 ± 0.2	2.00	83.2 ± 0.1
floored draw, no weights	20.3 ± 1.2	0.071 ± 0.003	75.6 ± 0.4	3.05 ± 0.07	82.6 ± 0.3
IW-GRESO (for reference)	12.0 ± 1.7	0.043 ± 0.002	74.2 ± 0.2	3.00 ± 0.10	82.8 ± 0.3
MoPPS	32.3 ± 0.7	0.306 ± 0.035	74.0 ± 3.5	13.18 ± 0.09	81.9 ± 1.6
+ visit cap $\Lambda_{\max} = 2.1$	26.0 ± 1.2	0.086 ± 0.003	77.7 ± 0.2	2.17	83.1 ± 0.2
+ cooldown 12	19.0 ± 1.2	0.082 ± 0.003	80.2 ± 0.8	2.00	82.7 ± 0.3
+ visit-triggered audit (verifier)	17.0 ± 0.6	0.034 ± 0.002	81.8 ± 0.4	13.39 ± 0.07	82.8 ± 0.1
Uniform, pool of 400	18.6 ± 2.9	0.059 ± 0.002	80.0 ± 0.8	1.00	81.5 ± 0.5
Uniform, pool of 800	11.3 ± 0.3	0.040 ± 0.002	72.8 ± 0.3	1.00	82.4 ± 0.8
Uniform, pool of 1,600	7.0 ± 0.9	0.029 ± 0.001	64.0 ± 1.0	1.00	82.6 ± 0.4

Table 37: Reward-hacking events on $\mathcal{H}$ against the number expected by chance (Math, Qwen3-1.7B-Base, pool of 800); mean ± SE over three seeds.

Arm	Events	Expected by chance	Chance-adjusted events
Uniform	11.3 ± 0.3	10.8 ± 0.2	$+0.6 \pm 0.1$
Dynamic sampling	12.0 ± 1.5	11.4 ± 1.4	$+0.6 \pm 0.2$
GRESO	29.8 ± 0.2	19.1 ± 0.8	$+10.8 \pm 0.6$
MoPPS	32.3 ± 0.7	13.9 ± 1.1	$+18.5 \pm 0.7$
DPPO	12.0 ± 1.2	11.3 ± 1.1	$+0.7 \pm 0.1$
Noise-corrected GRESO	28.3 ± 3.0	18.5 ± 2.1	$+9.8 \pm 0.9$
Prompt Replay	19.0 ± 1.7	14.9 ± 1.3	$+4.1 \pm 0.5$
IW-GRESO	12.0 ± 1.7	11.3 ± 1.6	$+0.7 \pm 0.2$
IW-MoPPS	13.0 ± 1.7	12.1 ± 1.6	$+0.9 \pm 0.1$
GRESO, floored draw without weights	20.3 ± 1.2	15.9 ± 1.0	$+4.5 \pm 0.2$
GRESO, visit bound	15.0 ± 1.7	12.6 ± 1.4	$+2.4 \pm 0.3$
GRESO, visit bound and triggered verification	17.3 ± 1.5	16.4 ± 1.4	$+1.0 \pm 0.2$

Table 38: Math with a pool of 3,200 prompts (Qwen3-1.7B-Base, Gemma-3-12B-it, 300 steps); mean ± SE over three seeds. Adjusted: events minus the number expected by chance. Leads are paired differences from uniform sampling in points of strict mastery at step 200, in runs rewarded by the exact verifier or by the judge.

Arm	Events	By chance	Adjusted	Lead, verifier	Lead, judge	MATH (%)
Uniform	17.3 ± 0.7	16.9 ± 0.6	$+0.4 \pm 0.1$	–	–	81.9 ± 0.6
GRESO	27.3 ± 1.2	21.3 ± 1.0	$+6.1 \pm 0.4$	$+8.9 \pm 0.4$	$+4.4 \pm 0.4$	81.6 ± 0.6
IW-GRESO	18.0 ± 1.0	17.4 ± 1.0	$+0.6 \pm 0.1$	$+1.5 \pm 0.3$	$+0.8 \pm 0.2$	82.1 ± 0.4
GRESO + bound + verif.	19.3 ± 0.7	18.4 ± 0.6	$+0.9 \pm 0.1$	–	$+3.9 \pm 0.3$	82.0 ± 0.4

Table 39: Math with Qwen3-8B-Base (pool of 800, Gemma-3-12B-it, 300 steps); mean ± SE over three seeds. Leads as in Table 38.

Arm	Events	By chance	Adjusted	Lead, verifier	Lead, judge	MATH (%)
Uniform	9.3 ± 0.3	8.9 ± 0.2	$+0.4 \pm 0.1$	–	–	90.4 ± 0.4
GRESO	22.0 ± 1.5	14.3 ± 0.7	$+7.7 \pm 0.8$	$+11.2 \pm 0.5$	$+3.4 \pm 0.4$	90.1 ± 0.6
IW-GRESO	10.3 ± 0.7	9.7 ± 0.6	$+0.6 \pm 0.1$	$+1.8 \pm 0.3$	$+0.9 \pm 0.2$	90.6 ± 0.3
GRESO + bound + verif.	13.0 ± 0.6	12.2 ± 0.5	$+0.8 \pm 0.1$	–	$+3.1 \pm 0.4$	90.5 ± 0.3

Table 40: Further baselines; mean ± SE over three seeds, conventions as in Table 2. Science columns give MMLU-Pro STEM and visit amplification for both policies; Math columns give strict mastery of $\mathcal{H}$ at step 200 (exact-verifier run and judge run) and reward-hacking events for Qwen3-1.7B-Base.

	Science, 1.7B		Science, 8B	Math, 1.7B		
Method	STEM ↑	Λ_{vis} ↓	STEM ↑	Exact ↑	Judge ↑	Events ↓
Uniform	48.9 ± 0.4	1.08	68.6 ± 0.5	84.7 ± 0.6	72.8 ± 0.3	$\mathbf{11.3 \pm 0.3}$
DPPO	$\mathbf{-1.9 \pm 0.7}$	**1.08***	$\mathbf{-1.4 \pm 0.7}$	$+0.4 \pm 0.7$	-0.4 ± 0.6	12.0 ± 1.2
Noise-corr. GRESO	-9.2 ± 1.9	5.81 ± 0.19	-6.4 ± 0.7	$+13.6 \pm 0.4$	$+4.3 \pm 0.7$	28.3 ± 3.0
Prompt Replay	-3.5 ± 0.9	1.90 ± 0.04	-2.4 ± 0.9	$+6.5 \pm 0.8$	$+2.4 \pm 0.6$	19.0 ± 1.7

Table 41: Science compute and exposure; mean ± SE over three seeds. Informative/1k counts judge-informative groups per 1,000 rollouts; tokens are completion tokens generated per run (millions). Max. weighted exposure is the largest realized weighted exposure of any prompt relative to uniform sampling's expected share; i.i.d. uniform draws of the same size give 1.78.

	Qwen3-1.7B-Base			Qwen3-8B-Base	
Method	Informative/1k	Tokens (M)	Max. weighted exposure	Informative/1k	Tokens (M)
Uniform	67.0 ± 0.4	43.3 ± 0.2	–	58.8 ± 0.4	52.0 ± 0.2
Dynamic sampling	58.2 ± 0.5	66.8 ± 0.3	–	50.6 ± 0.4	79.5 ± 0.4
GRESO	93.3 ± 0.7	42.2 ± 0.4	–	84.0 ± 0.7	49.8 ± 0.3
MoPPS	91.6 ± 0.6	43.2 ± 0.3	–	82.6 ± 0.7	50.5 ± 0.3
DPPO	79.0 ± 1.0	25.6 ± 0.1	3.70 ± 0.21	68.0 ± 1.1	30.7 ± 0.1
Noise-corr. GRESO	92.6 ± 0.5	42.0 ± 0.4	–	83.5 ± 0.7	49.6 ± 0.3
Prompt Replay	74.7 ± 0.6	44.9 ± 0.2	–	65.0 ± 0.8	53.2 ± 0.3
IW-GRESO	78.1 ± 0.4	44.6 ± 0.3	2.00 ± 0.05	68.0 ± 0.7	53.0 ± 0.3
IW-MoPPS	78.6 ± 0.5	44.3 ± 0.3	1.90 ± 0.03	68.4 ± 0.6	52.8 ± 0.3

Table 42: Math exposure measurements (Qwen3-1.7B-Base, pool of 800); mean ± SE over three seeds. Error-prone shares are in %; ESS is the Kish effective sample size over judge-informative groups per 1,000 rollouts (for GRESO, the number of judge-informative groups). Gaps of weighted rows are weighted; the unweighted value is in parentheses.

Arm	Weighted $\hat{\mathcal{E}}$ share	Visit $\hat{\mathcal{E}}$ share	Max. weighted exp.	Gap δ	ESS/1k
Uniform	13.7 ± 0.1	13.7 ± 0.1	–	0.040 ± 0.002	32.8 ± 0.2
GRESO	–	28.6 ± 0.4	–	0.129 ± 0.005	73.0 ± 0.7
DPPO	–	–	4.10 ± 0.21	0.044 (0.052)	37.0 ± 1.7
Noise-corr. GRESO	–	–	–	0.127 ± 0.012	–
Prompt Replay	–	–	–	0.075 ± 0.009	–
IW-GRESO	13.9 ± 0.1	21.0 ± 0.4	2.20 ± 0.06	0.043 (0.058)	38.2 ± 0.3
IW-MoPPS	13.8 ± 0.1	–	2.10 ± 0.03	0.045 (0.061)	35.5 ± 0.2

Table 43: Math held-out suite (greedy decoding, 2,048 tokens); mean ± SE over three seeds. The uniform row gives accuracy in %; the other rows give the paired difference from uniform sampling in points.

Method	AMC 2023 (40)	Minerva Math (272)	OlympiadBench (675)
Uniform	48.3 ± 0.8	30.1 ± 0.4	38.5 ± 0.4
Dynamic sampling	$+0.8 \pm 2.2$	$+0.4 \pm 0.7$	$+0.3 \pm 0.5$
GRESO	-0.8 ± 3.3	-0.5 ± 1.2	-0.4 ± 1.1
MoPPS	-1.7 ± 2.2	-0.5 ± 1.1	-0.4 ± 1.0
DPPO	-1.7 ± 0.8	-0.2 ± 0.5	-0.2 ± 0.3
Noise-corr. GRESO	-0.8 ± 2.2	-0.6 ± 1.0	-0.4 ± 1.0
Prompt Replay	0.0 ± 1.4	0.0 ± 0.7	$+0.1 \pm 0.6$
IW-GRESO	$+0.8 \pm 1.7$	$+0.2 \pm 0.3$	$+0.3 \pm 0.4$
IW-MoPPS	0.0 ± 1.4	$+0.1 \pm 0.6$	$+0.2 \pm 0.4$

Table 44: Group size in Science (Qwen3-1.7B-Base, Gemma-4-12B rubric judge, 406-prompt pool, 32 prompts per step, 200 steps). MMLU-Pro STEM accuracy, its paired per-seed difference from uniform sampling at the same group size, and three diagnostics of selection. Mean ± SE over three seeds; differences are paired per seed; bold marks the best value in each accuracy column within a group size. The visit amplification of uniform sampling is 1.08 in every seed because each epoch is a random permutation of the pool, and that of dynamic sampling is 1.06 per acquired prompt in every seed.

G	Arm	STEM (%)	Δ vs uniform	Visit conc.	Λ_{vis}	Late zero-variance
8	Uniform	48.9 ± 0.4	–	0.106 ± 0.001	1.08	0.570 ± 0.006
	DS	$\mathbf{49.0 \pm 0.3}$	$\mathbf{+0.1 \pm 0.7}$	0.104 ± 0.001	1.06	0.690 ± 0.006
	GRESO	38.8 ± 2.6	-10.1 ± 2.1	0.358 ± 0.008	5.96 ± 0.22	0.365 ± 0.009
	MoPPS	37.9 ± 0.5	-11.0 ± 1.0	0.451 ± 0.006	9.16 ± 0.22	0.250 ± 0.006
16	Uniform	49.5 ± 0.4	–	0.101 ± 0.001	1.08	0.470 ± 0.006
	DS	$\mathbf{49.7 \pm 0.3}$	$\mathbf{+0.2 \pm 0.7}$	0.100 ± 0.001	1.06	0.590 ± 0.006
	GRESO	40.3 ± 2.4	-9.2 ± 2.0	0.330 ± 0.007	5.52 ± 0.22	0.210 ± 0.006
	MoPPS	37.2 ± 0.6	-12.3 ± 1.0	0.488 ± 0.006	10.11 ± 0.15	0.130 ± 0.006
32	Uniform	49.8 ± 0.4	–	0.100 ± 0.001	1.08	0.370 ± 0.006
	DS	$\mathbf{50.0 \pm 0.3}$	$\mathbf{+0.2 \pm 0.7}$	0.100 ± 0.001	1.06	0.500 ± 0.006
	GRESO	45.5 ± 1.8	-4.3 ± 1.4	0.209 ± 0.006	3.11 ± 0.11	0.110 ± 0.006
	MoPPS	36.5 ± 0.6	-13.3 ± 1.0	0.517 ± 0.006	10.70 ± 0.16	0.060 ± 0.006

Table 45: Group size in Math (Qwen3-1.7B-Base, Gemma-3-12B judge, 800-prompt pool, 32 prompts per step, 300 steps). Reward-hacking events on $\mathcal{H}$ per run, the judged-minus-true gap δ over the last 20% of steps, visit amplification and held-out MATH accuracy. At $G = 16$ an event requires the judge to accept at least 12 of 16 responses ($\lceil 0.75G \rceil$). Mean ± SE over three seeds (at $G = 8$ each selector's two runs in seeds 0 and 1 are averaged within the seed first); differences are paired per seed; bold marks the best value in each column within a group size (lower is better for events and gap).

G	Arm	Events	Δ events	Gap δ	Λ_{vis}	MATH held-out (%)
8	Uniform	$\mathbf{11.3 \pm 0.3}$	–	$\mathbf{0.040 \pm 0.002}$	1.00	$\mathbf{82.4 \pm 0.8}$
	GRESO	29.8 ± 0.2	$\mathbf{+18.5 \pm 0.3}$	0.129 ± 0.005	11.40 ± 0.10	81.5 ± 1.3
	MoPPS	32.3 ± 0.7	$+21.0 \pm 0.6$	0.306 ± 0.035	13.18 ± 0.09	81.9 ± 1.6
16	Uniform	$\mathbf{10.0 \pm 1.2}$	–	$\mathbf{0.038 \pm 0.002}$	1.00	$\mathbf{83.0 \pm 0.6}$
	GRESO	28.0 ± 2.1	$\mathbf{+18.0 \pm 1.5}$	0.121 ± 0.014	10.42 ± 0.13	82.9 ± 1.1
	MoPPS	35.0 ± 3.2	$+25.0 \pm 2.6$	0.340 ± 0.045	14.69 ± 0.18	82.9 ± 1.3

Table 46: Learning rate in Science (settings of Table 44, $G = 8$). MMLU-Pro STEM accuracy of uniform sampling and GRESO, their paired difference, and GRESO's visit amplification. Mean ± SE over three seeds; differences are paired per seed; bold marks the best value in each column.

Learning rate	Uniform (%)	GRESO (%)	Δ GRESO − uniform	GRESO Λ_{vis}
1×10^{-6}	45.3 ± 0.5	$\mathbf{39.9 \pm 2.0}$	$\mathbf{-5.4 \pm 1.6}$	3.49 ± 0.22
3×10^{-6}	$\mathbf{48.9 \pm 0.4}$	38.8 ± 2.6	-10.1 ± 2.1	5.96 ± 0.22
1×10^{-5}	47.1 ± 1.0	34.7 ± 3.3	-12.4 ± 2.3	7.11 ± 0.22

Table 47: Learning rate in Math (settings of Table 45, $G = 8$). Reward-hacking events on $\mathcal{H}$, judged-minus-true gap δ (last 20% of steps), strict mastery of $\mathcal{H}$ at step 200 in the run trained with the judge, and held-out MATH accuracy. Mean ± SE over three seeds; differences are paired per seed; bold marks the best value in each column within a learning rate (lower is better for events and gap; strict mastery, which favors arms that revisit, is not bolded).

LR	Arm	Events	Δ events	Gap δ	Mastery (%)	Δ mastery	MATH (%)
10^{-6}	Uniform	$\mathbf{7.0 \pm 1.2}$	–	$\mathbf{0.030 \pm 0.001}$	64.0 ± 0.4	–	$\mathbf{80.6 \pm 0.8}$
	GRESO	19.0 ± 1.5	$+12.0 \pm 1.0$	0.088 ± 0.010	67.2 ± 1.0	$+3.2 \pm 0.7$	80.3 ± 1.2
3×10^{-6}	Uniform	$\mathbf{11.3 \pm 0.3}$	–	$\mathbf{0.040 \pm 0.002}$	72.8 ± 0.3	–	$\mathbf{82.4 \pm 0.8}$
	GRESO	29.8 ± 0.2	$+18.5 \pm 0.3$	0.129 ± 0.005	76.7 ± 0.5	$+3.9 \pm 0.7$	81.5 ± 1.3
10^{-5}	Uniform	$\mathbf{15.0 \pm 1.7}$	–	$\mathbf{0.058 \pm 0.002}$	74.1 ± 0.4	–	$\mathbf{79.7 \pm 0.8}$
	GRESO	37.0 ± 2.1	$+22.0 \pm 1.5$	0.184 ± 0.020	77.2 ± 1.0	$+3.1 \pm 0.6$	78.9 ± 1.2

Table 48: Unfiltered Science pool (406 prompts drawn uniformly from 1,500 candidates; otherwise as in Table 44 with $G = 8$). The late zero-variance share covers steps 160–200. Mean ± SE over three seeds; differences are paired per seed; bold marks the best value in each accuracy column.

Arm	STEM (%)	Δ vs uniform	Visit conc.	Λ_{vis}	Late zero-variance
Uniform	$\mathbf{46.2 \pm 0.5}$	–	0.100 ± 0.001	1.08	0.450 ± 0.006
DS	45.7 ± 0.2	$\mathbf{-0.5 \pm 0.7}$	0.100 ± 0.001	1.06	0.550 ± 0.006
GRESO	37.6 ± 2.5	-8.5 ± 1.9	0.303 ± 0.005	4.63 ± 0.28	0.253 ± 0.009
GRESO + visit-triggered audit	41.9 ± 2.1	-4.2 ± 1.6	0.263 ± 0.005	4.80 ± 0.39	0.357 ± 0.009

Table 49: Safeguards without the overlong penalty (Science, Qwen3-1.7B-Base, Gemma-4-12B judge, $G = 8$, 200 steps). MMLU-Pro STEM accuracy, paired difference from uniform sampling under the same recipe, and the step at which the runaway growth of response length began in seeds 0, 1 and 2. Mean ± SE over three seeds; differences are paired per seed; bold marks the best value in each column.

Arm	STEM (%)	Δ vs uniform	Runaway onset (seeds 0/1/2)
Uniform	$\mathbf{49.6 \pm 0.3}$	–	149 / 162 / 141
GRESO + visit-triggered audit	45.5 ± 0.8	$\mathbf{-4.1 \pm 1.1}$	none / none / 181
GRESO + visit cap 2.1	43.5 ± 1.1	-6.1 ± 1.4	91 / 97 / 86
Uniform + Rubric Dropout (per response)	34.6 ± 1.4	-15.0 ± 1.6	119 / 131 / 112

Table 50: Datasets and models, their role in this paper and their licenses or terms.

Asset	Role	License or terms
MATH (Hendrycks et al., 2021)	Math training pool (800 prompts) and held-out set (500 prompts)	MIT (original release)
RaR-Science (Gunjal et al., 2026)	Science training pool (406 prompts), held-out rubric set (300 prompts)	None stated; sources below
SCP-116K (Lu et al., 2025)	source of RaR-Science questions	CC-BY-NC-SA-4.0
NaturalReasoning (Yuan et al., 2025)	source of RaR-Science questions	CC-BY-NC-4.0
GeneralThought-430K	source of RaR-Science questions	MIT
MMLU-Pro (Wang et al., 2024)	Science transfer evaluation (physics, chemistry and biology, 3,148 items)	MIT
GPQA-Diamond (Rein et al., 2024)	Science transfer evaluation (8B policy)	CC-BY-4.0
Qwen3-1.7B-Base, Qwen3-8B-Base (Yang et al., 2025)	policies	Apache-2.0
Gemma-3-12B, Gemma-3-27B (Gemma Team et al., 2025)	Math training judge; comparison judge	Gemma Terms of Use
Gemma-4-12B, Gemma-4-26B-A4B (Gemma Team et al., 2026)	Science training judge; judge-strength judge	Apache-2.0
DeepSeek-V4-Flash (DeepSeek-AI, 2026)	reference judge; reward in the reference-as-reward runs	MIT (open weights), used via API
GLM-5.3-Flash (GLM-5 Team et al., 2026)	reference panel for Science; auditing judge in one Math audit variant	MIT (open weights), used via API
Kimi-K2.6 (Kimi Team et al., 2025; 2026)	reference panel for Science	provider terms, used via API