

Why Additive Process Rewards Wash Out in Group-Normalized Reinforcement Learning — and What It Takes to Measure It

Anonymous submission

Abstract

Verifiable process rewards—dense signals from checking intermediate reasoning steps—are widely expected to improve reinforcement learning of language-model reasoning. We show that in the standard group-normalized policy-optimization pipeline (GRPO), trajectory-additive process rewards are suppressed, in two regimes. In groups whose outcomes agree, the standardizing coefficient cancels exactly—an analytic consequence of the group-standard-deviation algebra—making the process term’s magnitude un-tunable, so coefficient sweeps are binary by construction. In groups whose outcomes disagree, the process term is compressed to a few percent of the advantage—empirically, a median of 0.3–6% for process scores whose within-group spread is small relative to the outcome spread, as in our setting. Measuring this requires a verifier fast and deterministic enough to instrument every reward call: we use a compact machine-checkable trace language whose rule-based linter catches 99.5% of injected rule-family reasoning errors (vs. 66.3% for self-verification and 87.2% for a frontier LLM judge) at 9 ms per trace with no model in the loop—an operating point complementary to proof assistants, which verify formalized proofs rather than native traces. In a pre-registered 12-run sweep (4B model, mathematical reasoning), the mechanism’s predictions hold and endpoint effects are null within a ~ 3 pp detection floor. We derive and test two pre-registered fixes and an external-result arbitration; a separately pre-registered endpoint candidate that cleared multiple-comparison correction did not survive fresh-seed replication and a magnitude-matched semantic control.

1 Introduction

Reinforcement learning with verifiable rewards has become the dominant recipe for improving the reasoning of language models, and a natural extension is to reward not just the final answer but the intermediate steps: if a checker can verify each line of a derivation, why not shape the policy toward derivations that check out? This intuition motivates a large and active body of process-reward and step-verification work. We report a mechanism that undercuts the simplest version of it, an instrument built to measure that mechanism, and a program of pre-registered experiments, one of which shows concretely what “measuring” an endpoint effect requires.

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

The mechanism is a property of group-relative policy optimization (GRPO), the standard critic-free algorithm, and specifically of its per-group reward standardization. When the reward is a trajectory-additive composite of a binary outcome term and a bounded process term, the process term’s influence on the policy gradient splits into two regimes with opposite pathologies. In sampling groups whose completions disagree on the outcome, the outcome variance dominates the group’s standard deviation and the process term is compressed to a few percent of the standardized advantage (a median of 0.3 to 6% across our sweep). In groups whose completions agree on the outcome, the standardizing denominator becomes the process term’s own spread, the coefficient cancels exactly, and the term’s magnitude becomes untunable, so a coefficient sweep is a binary experiment in disguise. Neither regime is a bug in an implementation; both follow from the standardization algebra, and together they retrodict a clean null we measured earlier and predict what happens when the standardization is changed.

Measuring any of this requires a verifier fast and deterministic enough to score every completion at every reward call — tens of thousands of line-level checks per run — which rules out both the learned process-reward models that dominate the literature (they are themselves models, and slow) and the formal proof assistants that anchor its rigorous end (they verify formalized statements behind an autoformalization barrier, at seconds-to-minutes latency). We use a compact, machine-checkable trace language whose rule-based linter verifies a native reasoning trace in about nine milliseconds with no model in the loop, detecting 99.5% of injected reasoning errors against 66.3% for self-verification and 87.2% for a frontier LLM judge. The linter is not the contribution; it is the instrument that makes the mechanism visible, and its operating point — deterministic verification of native traces, inside the reward loop — is complementary to, not competitive with, the proof-assistant end of the spectrum.

The paper makes three contributions. The two-regime washout mechanism comes with quantitative predictions that we confirm out of sample: disabling the standardizing divisor restores dose-responsive process influence across a coefficient ladder (six of six seed observations inside pre-registered bands), while an alternative token-level injection path breaks accuracy outright in a fully attributable way, which marks centering and outcome-gating as the load-bearing design

choices. The instrument and its detection study are the second contribution, situating deterministic native-trace verification against proof assistants and learned reward models. The methodological core is a pre-registered endpoint experiment in which the single surviving candidate — significant after multiple-comparison correction ($p=0.006$) and consistent with the mechanism — failed to replicate on fresh seeds and failed a magnitude-preserving, semantics-destroying control whose verdict we had committed to in advance. At the scale we study (a 4B model on mathematical reasoning), no process-reward configuration shows an endpoint benefit that survives replication and a semantic control, in either injection path. The episode answers the question in our title: measuring a verifiable-reasoning reward’s effect takes more than a clean gate; it takes replication and a control that separates what the verifier *measures* from the reward mass it *adds*.

2 Background and Related Work

Reasoning with and about intermediate steps. Chain-of-thought prompting established that eliciting intermediate steps improves mathematical reasoning (Wei et al. 2022), on benchmarks that remain the field’s yardsticks (Cobbe et al. 2021; Hendrycks et al. 2021). Reinforcement learning with verifiable rewards then made the *final answer* the training signal: group-relative policy optimization (GRPO) replaced the learned critic with within-group reward standardization (Shao et al. 2024), and scaled outcome-only training to strong reasoning models (DeepSeek-AI 2025). Our setting is exactly this pipeline, with one addition, a process term in the reward, and our subject is what the pipeline itself does to that addition.

What group normalization does. A rapidly consolidating line analyzes GRPO’s normalization for pure outcome rewards: removing the standard-deviation division to correct a difficulty bias (Liu et al. 2025), decoupled clipping and dynamic sampling (Yu et al. 2025), curvature accounts of why dividing by the group deviation helps (Ge et al. 2026), pathologies of the z -score under low-dispersion rewards (Salmani-Zarchi et al. 2026), corrections to the group-mean baseline (Garg et al. 2025), methods for exploiting the zero-variance groups that vanilla GRPO discards (Le et al. 2025), and a unification showing the major variants act on a single quantity, the group standard deviation (Bay and Yearick 2026). All of these treat rewards as given, and binary. To our knowledge none analyzes the *composite* case (an additive process term riding on the outcome reward), where, as Section 4 shows, the same denominator that stabilizes outcome learning silently reallocates and caps the shaping signal.

Process rewards and fine-grained credit. Process supervision at the step or token level is an active fix-space: aligning process with outcome rewards in policy optimization (Ding et al. 2026), generative credit assignment (Xie et al. 2025), temporal-difference-style credit in GRPO (Parthasarathi et al. 2025), outcome-grounded advantage reshaping (Li et al. 2026), and token-level reward models (Chen et al. 2025).

These works propose injection schemes. Contemporaneously, PASS (Wang et al. 2026a) observes that a composite reward’s process channel can be contaminated under group-relative optimization; our contribution is distinct and complementary along four axes — a two-regime *measurement* of the advantage displacement, the *analytic* cancellation of the additive coefficient in zero-outcome-variance groups, a *deterministic* verifier fast enough to instrument every reward call, and a semantic-shuffle-plus-fresh-seed *discipline* for endpoint claims — and Section 6 uses the measurement to make part of this design space testable at the advantage level (our confirmed predictions there are advantage-level, not endpoint-level). Closest to our arbitration experiment, a deterministic rule-based step verifier has been reported to improve a domain-specific reasoning task as an additive reward under this same family of pipelines (Pronesti, Belz, and Hou 2026); our mechanism’s simplest reading places that result in the regime where the aggregate process term is large relative to the outcome term, and we probe the reconciliation directly. Reward-hacking analyses motivate our monitoring choices (Wang et al. 2026b).

Verifying reasoning. Approaches to checking model reasoning range from formal proof assistants, which certify formalized statements at seconds-to-minutes latency behind an autoformalization barrier, through learned process-reward models and LLM judges, to deterministic symbolic checkers applied at evaluation time (He et al. 2026; Su 2025; Zhou et al. 2026). Section 3 argues these are operating points on a latency–coverage–trust frontier, and that a deterministic, model-free checker over native traces is the operating point the reward loop of reinforcement learning can afford. Efficient- reasoning work (length penalties, budget control, the accuracy-per-token trade-off; Sui et al. 2025; Dumitru et al. 2025; Yi and Wang 2025; Wen et al. 2025; Zhang et al. 2026; Gao et al. 2025) is adjacent but orthogonal here: we use token budgets only as evaluation protocol, and make no efficiency claim in this paper.

3 The Instrument: A Deterministic Trace Verifier

Measuring Section 4’s quantities requires scoring the process quality of every completion at every reward call—and doing so deterministically, because a verifier whose own judgments are noisy or model-dependent cannot serve as the measurement standard for an analysis of reward noise. We use Telegraph Reasoning (TE), a compact trace language in which the model states givens, goals, tagged equations, and explicit checks in a fixed line grammar, terminating in a machine-parsable answer line. A rule-based linter enforces the grammar and, for each equation or check line, attempts deterministic verification by symbolic substitution: every line receives a verdict of *pass*, *fail*, or *uncheckable*, and the trace as a whole a structural validity bit. Twelve rules cover structure, scoping, and numeric consistency; the implementation is pure rules plus a computer-algebra call, with no language model anywhere in the loop, and verifies a trace in 9 ms at the median. The full grammar and rule specification appear in the supplementary material.

Verifier	Injected errors			Organic traces valid-rate span
	TPR	FPR	Loc.±3	
TE linter (ours)	99.5	10.0	90.3	21.8–27.2
Self-verify (NP)	66.3	—	—	n/a
LLM judge (frontier)	87.2	—	—	n/a

Table 1: Discriminative power of the instrument (Section 3). Injected-error columns from the corruption benchmark (five corruption types; localization = fraction of detections within ± 3 lines); the benchmark favors the linter by construction (shared rule vocabulary), so the final column reports discrimination on *organic* RL outputs: full-trace validity span across the twelve γ -sweep cells.

Why deterministic: discriminative power. On a corruption benchmark of reasoning traces with injected errors spanning five corruption types, the linter detects 99.5% of errors at a 10% false-positive rate; a self-verification protocol in the Natural Program style detects 66.3%, and a frontier LLM judge prompted for the same task 87.2%. Two limitations qualify this comparison. The detection rates compare true-positive rates that are not false-positive-matched across the three verifiers, so the headline gap overstates a like-for-like advantage; and the injected errors are drawn from the same failure vocabulary the linter’s rules encode, so the benchmark is favorable to it. Two observations temper the concern. First, the same study measures localization—90.3% of detections within three lines of the injected fault—which a vocabulary-matched but shallow checker would not achieve. Second, and more directly, the linter discriminates on *organic* model output, where no injection occurred: across our twelve reinforcement-learning runs its full-trace validity rate spans 21.8–27.2% by cell on the evaluation set, moves measurably over training on held-out problems (by -2.0 to $+8.0$ pp per run), and its per-line verdicts generate the group statistics of Section 4—structure that could not arise from a checker that merely echoed its own injections.

The operating point: verification inside the reward loop. Formal proof assistants occupy one end of the verification spectrum: they certify complete correctness, but of *formalized* statements, and reaching them from natural mathematical text requires autoformalization, which remains unreliable precisely on the informal reasoning reinforcement learning must score, with verification latencies of seconds to minutes per attempt (DeepSeek-AI 2025; Pronesti, Belz, and Hou 2026). Learned process-reward models occupy the other end: they score native text directly, but are themselves models—opaque, biased toward surface features, and attackable by the very policy they train (Wang et al. 2026b). The linter sits at a third point: verification of *native* traces, deterministic, and cheap enough to be invisible in the training budget. A single training run here makes roughly 9×10^3 full-trace verifications (6,000 in the reward path, 3,000 on held-out checkpoints) comprising some 4×10^4 line-level checks; at 9 ms per trace this is well under 0.1% of a ~ 30 -second optimization step, where a seconds-scale verifier would consume 10–40% of it and an autoformalization round-trip is infeasible

in-loop. The costs are complementary rather than competitive: the linter certifies much less than a proof assistant, checking line-local consistency rather than theoremhood, but it is a practical operating point that can sit inside the reward loop and stamp every completion the policy produces, which is what the measurement in this paper requires. The trace language restricts what the model may write, and we measure rather than assume the cost of that restriction: expressibility coverage of the training distribution is reported in the supplementary material.

4 The Mechanism: Two Regimes of Washout

Group-relative policy optimization draws G completions per prompt, scores each with a scalar reward, and converts rewards to advantages by standardizing within the group: $A_i = (R_i - \bar{R})/(\sigma_R + \epsilon)$, with σ_R the within-group sample standard deviation and $\epsilon = 10^{-4}$ (Shao et al. 2024). When the reward is a trajectory-additive composite $R_i = r_{\text{ans},i} + \gamma s_i$, where $r_{\text{ans}} \in \{0, 1\}$ is the verifiable outcome and s_i is a bounded process score, the fate of the process term is decided entirely by the composition of the group, and it splits into two regimes with opposite pathologies.

Regime 1: outcome variance present. If the group contains both correct and incorrect completions, σ_R is dominated by the outcome spread. In our training runs the within-group standard deviation of r_{ans} averages 0.19, while that of γs reaches only 0.002 at $\gamma=0.1$ and 0.021 at $\gamma=1.0$ with $\beta=0.1$ process scoring (Section 5). Restricted to groups with a well-conditioned denominator ($\sigma_R \geq 0.05$), the process term displaces post-normalization advantages by a median of 0.3–6% across γ , against outcome-driven displacements on the order of one unit (the full per- γ series is in the supplement). This is an empirical magnitude bound conditional on $\sigma(\gamma s) \ll \sigma(r_{\text{ans}})$ —a fact about our process-score construction, not the standardization algebra—measured at one scale, group size ($G=4$), and model class; it is not an impossibility theorem.

Regime 2: outcome variance absent. If every completion in the group receives the same outcome score—all wrong, or all right—the outcome contributes nothing to either the numerator or the denominator, and the advantage reduces to $A_i = \gamma(s_i - \bar{s})/(\gamma\sigma_s + \epsilon)$. For $\gamma\sigma_s \gg \epsilon$ the coefficient cancels:

Lemma 1. *In a group with constant outcome reward and non-degenerate process scores, group-standardized advantages are invariant to the process coefficient: $A_i \rightarrow (s_i - \bar{s})/\sigma_s$ as $\gamma\sigma_s/\epsilon \rightarrow \infty$, independent of γ .*

The lemma is an immediate consequence of the shared-denominator algebra that recent analyses have used to unify GRPO variants (Bay and Yearick 2026), and we present it as such rather than as a novelty; its practical force in the composite-reward setting appears to be unexamined. In these groups the process term is not suppressed at all: it is the *entire* standardized advantage, at full magnitude, but its strength is untunable, since any $\gamma > 0$ produces the same advantages as any other.

The rescue channel and its saturation. Regime 2 is not a corner case. At $\gamma = 0$, 47.8% of training groups produce identically zero advantage (no gradient from the prompt at that step), and 18.0% of all groups — roughly 38% of the zero-variance ones — carry non-degenerate process scores that could differentiate their completions. Introducing any positive coefficient activates every such group and then saturates: the zero-advantage fraction drops to $\approx 32\%$ at $\gamma = 0.1$ and does not move further as γ grows to 1.0 (a matching saturation appears in per-group ranking statistics, reported in the supplement). The consequence for experiment design is a scoped one: because the live rescue channel saturates at any $\gamma > 0$ while the Regime-1 scaling channel remains but is bounded at the measured displacement share, a coefficient sweep of a trajectory-additive process reward behaves as an approximately binary treatment ($1[\gamma > 0]$) rather than a dose. The two regimes are asymmetric in status: Regime 2 is analytic (Lemma 1, a corollary of the shared-denominator algebra), while Regime 1 is a purely empirical magnitude observation. This picture retrodicts the shape of our endpoint results (Section 5)—no dose-response was constructible—and generates the out-of-sample predictions of Section 6: removing the standard-deviation division should restore β -linear process influence in all groups (prediction D1); injecting credit at the token level bypasses the scalar bottleneck entirely (D3); and additive composites in which the aggregate process term is *large* relative to the outcome term—the regime of a recently reported positive result with a deterministic step verifier (Pronesti, Belz, and Hou 2026)—should escape Regime 1’s bound (D4).

These within-group statistics are logged at every reward call (Section 3 describes the verifier that makes this practical) and are stationary across training. One measurement caveat is load-bearing: the ϵ -regularized denominator inflates naive means over displacement ratios by 20–30 $\times$ through a small- σ_R tail, so we report medians and restricted-denominator variants throughout.

5 A Pre-Registered Null, Explained

A note on notation before the experiments. Two coefficients appear: γ scales the entire process bundle s in the trajectory reward, while β scales only the line-credit term inside s . The sweep of Section 5 varies γ at fixed $\beta=0.1$; the probe of Section 6 instead fixes $\gamma=1$ and varies β , so “the process weight” there refers to β . Where Section 4’s cancellation is at issue, it is γ (the additive-bundle coefficient) that cancels, independent of β . Because β leaves the uncheckability penalty fixed while γ scales the complete bundle, the two sweeps test related but non-equivalent interventions.

We trained twelve policies— $\gamma \in \{0, 0.1, 0.5, 1.0\}$, three seeds each—on a shared pool of mathematics problems disjoint from all evaluation sets, under a single hardware and kernel configuration, with the $\gamma = 0$ cells re-run inside the sweep so that the outcome-only baseline shares every execution detail with the treatment arms. Concretely: a 4B model (Qwen3.5-4B) with QLoRA adapters, group size $G=4$, 1,500 optimization steps per run, greedy decoding, evaluated at generation caps $\{256, 512\}$ and full length. The process score $s = \beta r_{\text{line}} - p_{\text{uncheck}}$ bundles the linter’s per-line pass rate

γ	Accuracy mean $\pm$ sd	Lint-valid mean $\pm$ sd	Verif.- corr.	$\Delta\text{Acc vs } \gamma=0$ paired [95% CI]
0.0	37.07 $\pm$ 0.90	23.53 $\pm$ 1.45	13.93	—
0.1	37.67 $\pm$ 0.76	25.73 $\pm$ 1.29	14.73	+0.60 [−1.20, +2.40]
0.5	36.93 $\pm$ 0.70	25.47 $\pm$ 1.10	14.33	−0.13 [−1.87, +1.67]
1.0	38.53 $\pm$ 0.31	23.67 $\pm$ 2.01	14.00	+1.47 [−0.27, +3.20]

Table 2: The γ -sweep endpoints (Section 5). Twelve runs, uniform configuration; $\gamma=0$ is the outcome-only baseline re-run inside the sweep. No paired contrast excludes zero (largest: $\gamma=1.0$, McNemar $p=0.115$); the lint-valid low- γ bump does not survive family-wise correction (Section 5). Verified-correct = answer correct *and* trace linter-valid.

with an uncheckability penalty at fixed $\beta = 0.1$; $\gamma = 0$ reproduces the outcome-only reward bit-for-bit, a property locked by unit test rather than assumed. All twelve cells were evaluated greedily on the same 500 MATH-500 problems in identical order, enabling paired-by-problem inference; the gate criteria, sweep grid, and diagnostics were fixed before any training run.

Accuracy shows no effect at any coefficient (Table 2). The largest contrast, $\gamma = 1.0$ against $\gamma = 0$, is +1.47pp with a paired 95% confidence interval of [−0.27, +3.20] and McNemar $p = 0.115$ over 1,500 paired problem-seed observations. This pooled problem-seed test is conditional on the realized policies: it treats each trained checkpoint as fixed and does not estimate the additional uncertainty over training seeds, so its intervals are narrower than a seed-random-effects analysis would give (a limitation we return to for the confirmation stage). The per-problem dose-response slope is +1.11pp per unit γ with an interval crossing zero, and a label-permutation test over the twelve cell means gives $p = 0.095$. Linter-validity, the metric the process term directly rewards, shows a suggestive low-coefficient anomaly (+2.20pp at $\gamma=0.1$, uncorrected $p = 0.014$; +1.93pp at $\gamma=0.5$) that is absent at $\gamma = 1.0$, survives Holm correction only within its own three-test family, dies across the nine-test primary family, and vanishes under the pooled any- γ -positive contrast (+1.42pp, CI [−0.07, +2.93]). Verified-correct rate (answer correct *and* trace linter-valid) is null everywhere. Across the seventeen tests this analysis reports, roughly 0.8 would be expected below $p = 0.05$ under a global null; exactly two were, both in the linter-validity family just described. For the program as a whole, we evaluated two pre-registered endpoint gates (this γ -sweep and the coefficient/injection probe of Section 6); the single candidate that cleared either was carried to a dedicated confirmation stage whose analysis was frozen before it ran (Section 6), so the search over configurations that produced it is handled by replication rather than by a family-wise correction over the whole program.

Two qualifications are part of the claim rather than caveats to it. First, the design is powered for effects of roughly three percentage points and cannot exclude smaller ones; the $\gamma = 1.0$ accuracy lean (positive in three of three seeds) is unproven, not disproven. Second, held-out trajectories during training close the remaining escape route: linter compli-

ance on a 200-problem held-out slice rose by +4.0pp in the *outcome-only* arm (as much as in any process arm, +1.8 to +3.5pp), and the correlation between γ and held-out linter rise across the twelve runs is $r = 0.04$ ($p = 0.89$). Whatever trace-formatting drift reinforcement learning induces here is a property of optimizing for answers, not of the process term. A reward-hacking axis evaluated per checkpoint (rising linter compliance against flat held-out accuracy) triggered in none of the twelve runs.

The null survives adversarial scrutiny by construction rather than by assertion: a cross-model integrity audit (two reviewer models from different families) verified ground-truth provenance, statistics, and the extraction pipeline, and its findings (an extraction-completeness guarantee retrofitted with hash-level verification, an overclaim in our own mechanism wording that we retracted, and denominator-artifact hygiene in the diagnostics) are folded into every number quoted here. Combined with Section 4, the result is a null with a cause: the diagnostics show the process signal being suppressed in one regime and saturated in the other, the endpoints show the consequence, and Section 6 tests whether the same account predicts what happens when the suppression is removed.

6 Predictions Out of Sample

Section 4’s account was frozen, with quantitative bands, before a second pre-registered experiment: seven configurations probing the two injection paths the mechanism identifies, at fixed model, data, and optimization settings (fourteen runs; configurations and predictions were committed to hash-identified manifests before any run started).

D1: removing the divisor restores dose response. The mechanism predicts that disabling the standard-deviation division (*scale_rewards=none*, retaining group mean-centering) makes the process term’s advantage share β -linear everywhere, within a factor of two of $\beta/0.1 \times 0.0344$ (the Regime-1 reference measured in Section 4). Measured (robust medians, per seed): 0.043/0.034 at $\beta=0.1$ against band [0.017, 0.069]; 0.145/0.150 at $\beta=0.5$ against [0.086, 0.344]; 0.206/0.237 at $\beta=1.0$ against [0.172, 0.688], with six of six seeds in band and the $\gamma=0$ counterfactual at exactly zero in both its seeds. The shares run 1:3.8:5.7 against β ratios of 1:5:10: β -monotone, but sub-linear, the largest cell falling nearly $2\times$ short of a strictly linear extrapolation. We read this as self-damping (a large process term inflates the group spread it is normalized against), but we flag that this is a *post-hoc* reading, not part of the frozen prediction, which committed only to the factor-of-two bands; the bands are deliberately loose and we report six cell-seeds (three β cells $\times$ two seeds, not six independent draws) landing inside them, with the caveat that a tighter linear band would have flagged the $\beta=1.0$ point. The companion prediction D2 also held: with the divisor gone, uniformly-wrong groups carry β -proportional advantages (0.048/0.127/0.208 median maxima) rather than the saturated $\pm\approx 1$ that standardization manufactured, and exactly zero at $\gamma=0$, where outcome-only training gives such groups no gradient at all.

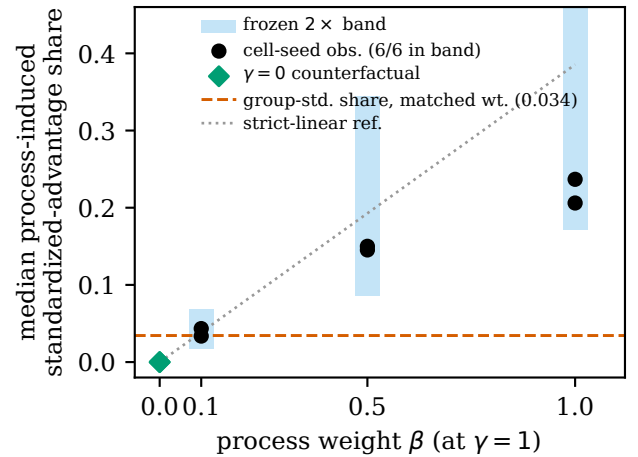


Figure 1: D1: disabling the standardizing divisor restores dose-responsive process influence. Process-term advantage share (robust median over groups with $\sigma_R \geq 0.05$) versus the process weight β at $\gamma=1$, under *scale_rewards=none*. Points are six cell-seed observations (two seed identities); shaded regions are the factor-of-two bands frozen before the runs, and 6/6 observations fall inside them. The green diamond is the $\gamma=0$ counterfactual (share exactly zero; outcome-only training gives these groups no gradient). The orange dashed line marks the share measured *under* group standardization at the matched setting — the earlier sweep’s maximum tested effective line-credit weight ($\gamma=1.0$, $\gamma\beta=0.1$; robust median 0.034). The unwashed $\beta=0.1$ share coincides with it and rises from there with β — weights beyond this were unreachable in the standardized sweep, whose bundle coefficient cancels in uniform-outcome groups (Lemma 1). The rise is monotone but sub-linear against the dotted strict-linear reference (the $\beta=1$ point falls short, a post-hoc self-damping reading; see text). The x -axis is linear so the counterfactual sits at its true $x=0$. Advantage-level prediction; endpoint improvement did not replicate (Section 6).

D4: a prediction that failed, reported as such. We pre-registered an arbitration of an external result: a deterministic step-verifier reward summed over many steps has been reported to improve a structured-reasoning task as an additive composite under this same normalization family (Pronesti, Belz, and Hou 2026), and our mechanism’s simplest reading — additive composites work when the aggregate process share is large — predicted that our largest-share cell ($\beta=1.0$, ≈ 21 – 24% advantage share) would show the endpoint effect smaller-share cells lack. It did not (verified-correct rate +0.7pp at the 256-token cap, $p=0.13$). The naive magnitude story is therefore incomplete: D1 and D2 confirm the mechanism’s *advantage-level* predictions, but D4 was its one *endpoint* prediction and it failed, so the mechanism is, at this point, confirmed as an account of the advantage displacement and unconfirmed as a predictor of endpoints. Two statuses should be kept distinct here. The pre-registered operationalization of D4 failed by its frozen rule. As an *arbitration of the external result*, however, the test is inconclusive rather

than negative: our $\beta=1.0$ cell differs from that setting in group size, reward aggregation (trajectory-level versus per-step-summed), and domain, and we did not run their configuration at our scale (a direct probe we flag as the natural next experiment). “Large aggregate share” is thus a hypothesis our indirect test did not bear out, not an explanation we can assert or refute. We keep the failed prediction in the record because the confirmed ones are only as credible as our willingness to report this one.

An endpoint candidate that did not survive its controls.

Restoring dose response produced one endpoint candidate. The $\beta=0.5$ unwashed-additive configuration cleared our pre-registered endpoint gate on the selection set (verified-correct rate at the 256-token cap $+1.90\text{pp}$, Holm-adjusted $p=0.006$, paired 95% CI $[+0.8, +3.0]$), and its mechanism signature was in band (D1 above). Under the confirmation protocol we committed to before running it, it did not survive, on two independent grounds. First, on fresh seeds the effect did not replicate ($+0.00\text{pp}$, McNemar $b=c=19$, $p=1.0$; full-generation accuracy leaned negative, -1.9pp , $p=0.078$): the fresh-seed failure alone rejects the candidate. Second, the within-group shuffled control, which preserves the process reward’s magnitude distribution per group while destroying its association with trace content, reproduced and slightly exceeded the candidate’s effect in every slice ($+0.70\text{pp}$ vs. P0 at the 256-token cap): whatever signal exists is not attributable to what the verifier measures. By the binding rule fixed in advance (ambiguity counts against the candidate), the candidate is rejected. We are careful about what the data do and do not establish: they reject P2 and reject a semantic attribution of its probe-stage gain, but they do not distinguish among seed selection, optimization noise, and a magnitude effect as the source of that gain, and we do not claim to. We report the episode prominently because it is the sharpest evidence for this paper’s thesis: a pre-registered, multiple-comparison-corrected, mechanism-consistent endpoint result did not survive fresh-seed replication and a semantic-association control. Measuring a verifiable-reasoning reward’s *effect* takes more than a clean single-experiment gate; it takes replication and a control that separates semantics from magnitude. The mechanism results (D1/D2) are unaffected: they are advantage-level claims, confirmed on their own pre-registered bands and independent of any endpoint outcome.

D3: where token-level credit breaks. The second injection path bypasses the scalar entirely: token-level advantage offsets from the linter’s per-line verdicts, in our *uncentered verifier-line token-offset* formulation — tokens inside pass/fail/uncheckable lines receive a constant signed offset, unnormalized and independent of the trajectory’s outcome. At low weight ($\gamma_{tok}=0.1$) this is a mild, harmless drift; at $\gamma_{tok}=0.5$ it is destructive, and the damage is fully attributable. Three independent instruments agree. The token-advantage diagnostics show 50–60% of completion tokens carrying offsets whose per-token distortion scales exactly with dose (token-advantage deviation $0.050 \rightarrow 0.245$ for a $5\times$ weight increase, about half the scale of the group-normalized scalar advantage, applied always-on). The emitted traces restructure accordingly: at $\gamma_{tok}=0.5$ the policy

writes $\sim 22\%$ fewer characters but $\sim 60\%$ more equation lines and $\sim 75\%$ more check lines, halving its untagged derivational prose. The reward has become an annuity on checkable-line tokens, paying every token unconditionally, where the outcome anchor pays once per trajectory, group-relative. And held-out accuracy declines *during training* in both seeds (-3.5 and -9.0pp first-to-last), so the harm is an optimization pathology accruing over steps, not an evaluation artifact; final accuracy lands 7.5pp below baseline. This is the hacking surface we pre-registered for this formulation (“verbose-but-checkable padding”), realized as checkable-line *density* rather than length, and caught in-flight by the same instrumentation that measured the mechanism. We scope the finding precisely: it falsifies *this* uncentered, outcome-decoupled token-offset design at meaningful weight, and marks centering and outcome-gating as the load-bearing choices any token-level verifier credit must get right — not token-level credit as a class.

7 Implications and Limitations

For practitioners adding a process reward to GRPO, the mechanism yields a concrete design rule: under per-group standardization, a trajectory-additive process term cannot be tuned into relevance by its coefficient — the coefficient cancels wherever the term is the only signal and is swamped wherever the outcome varies. If the process signal is to matter at all, the standardization must be changed (removing the divisor restores dose-response, as our D1 experiment confirms) or the signal injected off the scalar advantage entirely — and our token-level attempt shows that the latter is not free: an uncentered, outcome-decoupled per-token offset becomes an annuity on checkable tokens and degrades the answers it was meant to support. Centering and outcome-gating are the load-bearing choices for any token-level verifier credit; we did not find a setting of our formulation that helped.

The broader implication is methodological. Our surviving endpoint candidate was pre-registered, corrected for multiple comparisons, significant at $p=0.006$, and consistent with an independently confirmed mechanism — four properties usually taken as strong evidence — and it still failed to replicate on fresh seeds and failed a magnitude-matched semantic control. We argue the exposure is endemic, not incidental to our setup. Any process reward adds reward *mass* correlated with the process metric it optimizes; under group-relative advantage estimation, adding mass to a subset of completions shifts their advantages whether or not the mass tracks anything semantic, and the resulting endpoint movement is observationally identical to a genuine effect on the metric the reward and the evaluation share. This is a structural property of shaped rewards under group normalization, not a quirk of linter validity: any verifiable-reasoning reward whose target metric correlates with its own magnitude inherits it. The standard defenses do not address it: a held-out set, multiple seeds at train time, and multiple-comparison correction all leave a magnitude-driven gain indistinguishable from a semantic one, as our own $p=0.006$ candidate shows. The control that separates the two is a within-group permutation that preserves the reward’s magnitude distribution while destroying its association with trace content: if the permuted reward

reproduces the gain, the gain cannot be attributed to what the verifier measures. We propose this shuffle as a standard, near-zero-cost control for endpoint claims about verifiable process rewards, and we release its implementation so it can be dropped into an existing GRPO reward path.

Limitations bound these conclusions and we state them without hedging. All experiments use a single 4B model with QLoRA adapters on mathematical reasoning with a compact trace language; the mechanism’s algebra is general to group-standardized objectives, but its *magnitudes* — the specific few-percent displacement share — are measured at one scale, one group size, and one process-score scale, and larger models or different verifiers may sit at different operating points. Our endpoint statistics are powered for effects of roughly three percentage points and cannot exclude smaller ones; we claim a null within that floor, not a proof of no effect. The trace language restricts what the model may express, which bounds its accuracy ceiling independently of the reward question; we report expressibility coverage in the supplement but do not disentangle it here. And our arbitration of a positive external result with a deterministic step verifier remains open: our largest-share configuration did not reproduce that regime’s benefit, but the two settings differ in group size, per-step versus trajectory-level reward aggregation, and domain, so we report the prediction as failed rather than the external result as explained.

8 Reproducibility and Artifacts

Every experiment in this paper was pre-registered before it ran: the sweep grid, the coefficient bands for the mechanism predictions, the endpoint gate criterion with its multiple-comparison correction, and the confirmation protocol with its binding shuffle-control rule were all committed to hash-identified configuration manifests and frozen plan documents, with dated records, prior to the corresponding runs. The distinction between the selection set (on which all model and coefficient choices were made) and the confirm-stage analysis is preserved throughout, and the endpoint candidate’s death was decided by a rule fixed in advance of the data.

The supplementary material contains the trace-language grammar and the full rule specification of the linter; the reward and advantage-injection code for both fixes with the unit tests that lock their invariants (including the coefficient-zero equivalences and the within-group shuffle’s magnitude-preservation and group-locality); the per-cell diagnostic and eval records with the analysis scripts that regenerate every number in the paper; the audit reports; and the pre-registration documents with their revision history. All numbers reported here trace to those records through a zero-context claim audit run before submission. The supplement is self-contained (no external links) and anonymized.

We intend to publicly release the trace language, the linter, the advantage-injection implementations, and the instrumentation and analysis harness, so that both the mechanism measurements and the confirmation protocol can be reproduced and reused as controls for verifiable-reasoning reward claims.

References

- Bay, Y. Y.; and Yearick, K. A. 2026. GRPO, Dr. GRPO, and DAPO Are Three Operations on One Number: The Group-Standard-Deviation Identity. arXiv:2607.00152.
- Chen, H.; Yang, T.; Gao, S.; Chen, R.; Quan, X.; Tian, H.; and Yao, T. 2025. Discriminative Policy Optimization for Token-Level Reward Models. arXiv:2505.23363.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.
- DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.
- Ding, R.; Lv, Y.; Meng, X.; Song, J.; Wang, C.; Jiang, C.; and Cheng, Y. 2026. PRPO: Aligning Process Reward with Outcome Reward in Policy Optimization. arXiv:2601.07182.
- Dumitru, R.-G.; Peteleaza, D.; Yadav, V.; and Pan, L. 2025. ConciseRL: Conciseness-Guided Reinforcement Learning for Efficient Reasoning Models. arXiv:2505.17250.
- Gao, J.; Yan, S.; Tan, Q.; Yang, L.; Xu, S.; Fu, W.; Mei, Z.; Lyu, K.; and Wu, Y. 2025. How Far Are We from Optimal Reasoning Efficiency? arXiv:2506.07104.
- Garg, A.; Zhang, C.; Neema, N.; Bick, D.; Venkatesh, G.; and Hestness, J. 2025. CoRPO: Adding a Correctness Bias to GRPO Improves Generalization. arXiv:2511.04439.
- Ge, C.; Yin, C. H.; Liang, H.; and Zhang, J. 2026. Why GRPO Needs Normalization: A Local-Curvature Perspective on Adaptive Gradients. arXiv:2601.23135.
- He, P.; Huang, Y.; Sachan, M.; and Jin, Z. 2026. Uncovering Hidden Correctness in LLM Causal Reasoning via Symbolic Verification. arXiv:2601.21210.
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.
- Le, T.-L. V.; Jeon, M.; Vu, K.; Lai, V.; and Yang, E. 2025. No Prompt Left Behind: Exploiting Zero-Variance Prompts in LLM Reinforcement Learning via Entropy-Guided Advantage Shaping. arXiv:2509.21880.
- Li, Z.; Kang, L.; Xiao, F.; Xing, L.; Si, Q.; Li, Z.; Gong, W.; Yang, D.; Xiao, Y.; and Guo, H. 2026. Outcome-Grounded Advantage Reshaping for Fine-Grained Credit Assignment in Mathematical Reasoning. arXiv:2601.07408.
- Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding R1-Zero-Like Training: A Critical Perspective. arXiv:2503.20783.
- Parthasarathi, P.; Reymond, M.; Chen, B.; Cui, Y.; and Chandar, S. 2025. GRPO- λ : Credit Assignment improves LLM Reasoning. arXiv:2510.00194.
- Pronesti, M.; Belz, A.; and Hou, Y. 2026. Beyond Outcome Verification: Verifiable Process Reward Models for Structured Reasoning. arXiv:2601.17223.

Salmani-Zarchi, M. M.; Rahimi, Z.; Faili, H.; and Dousti, M. 2026. MDP-GRPO: Stabilized Group Relative Policy Optimization for Multi-Constraint Instruction Following. arXiv:2606.06058.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.-M.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Su, C. 2025. MedRule-KG: A Knowledge-Graph-Steered Scaffold for Mathematical Reasoning with a Lightweight Verifier. arXiv:2510.16309.

Sui, Y.; Chuang, Y.-N.; Wang, G.; Zhang, J.; Zhang, T.; Yuan, J.; Liu, H.; Wen, A.; Zhong, S.; Chen, H.; and Hu, X. 2025. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. arXiv:2503.16419.

Wang, C.; Tian, H.; Yang, T.; Shi, Y.; Yao, T.; and Ding, W. 2026a. Process Advantage Signal Shaping: A Paradigm-Agnostic Middleware for Process-Supervised RL in LLM Reasoners. arXiv:2606.29296.

Wang, S.; Pham, Q. H.; Yin, F.; Wang, X.; Chen, J. Q.; Durrett, G.; and Ye, X. 2026b. Detecting and Suppressing Reward Hacking with Gradient Fingerprints. arXiv:2604.16242.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E. H.; Xia, F.; Le, Q.; and Zhou, D. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wen, H.; Wu, X.; Sun, Y.; Zhang, F.; Chen, L.; Wang, J.; Liu, Y.; Liu, Y.; Zhang, Y.-Q.; and Li, Y. 2025. BudgetThinker: Empowering Budget-aware LLM Reasoning with Control Tokens. arXiv:2508.17196.

Xie, G.; Shi, Y.; Tian, H.; Yao, T.; and Zhang, X. 2025. CAPO: Towards Enhancing LLM Reasoning through Generative Credit Assignment. arXiv:2508.02298.

Yi, J.; and Wang, J. 2025. ShorterBetter: Guiding Reasoning Models to Find Optimal Inference Length for Efficient Reasoning. arXiv:2504.21370.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; et al. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. arXiv:2503.14476.

Zhang, Q.; Guo, T.; Ren, X.; Chen, J.; Ding, M.; Xin, R.; and Xiao, X. 2026. Scaling Reasoning Tokens via RL and Parallel Thinking: Evidence From Competitive Programming. arXiv:2604.01302.

Zhou, H.; Yang, A. X.; Aitchison, L.; Korhonen, A.; and Jiang, A. Q. 2026. Reasoning Arena: Trace Tournaments When Verifiable Rewards Fall Short. arXiv:2606.09380.