Sign-Safe Advantage Composition for Multi-Objective RLVR

Anonymous ACL submission

Abstract

Multi-objective reinforcement learning with verifiable rewards (RLVR) combines binary outcome verifiers with continuous auxiliary signals via additive composition. We show that this has a failure mode: a non-trivial fraction of rollouts have advantages with sign opposite to the verifier's judgment, causing the policy gradient to push the model toward verifier-rejected outputs. We find that 8-39% of advantages are sign-disagreeing across 3 tasks. We find unstable training in such regimes, including training collapse across multiple seeds on MATH with PRM guidance. We propose Sign-Safe Advantage Composition (SSAC), an advantage construction that bounds the auxiliary's per-rollout contribution below the outcome anchor magnitude, guaranteeing sign agreement with the verifier on every rollout. SSAC eliminates sign-disagreement, mitigating training collapse and outperforming outcome-only GRPO on the MATH Level 5 subset of problems by +12.5 percentage points and all aggregation methods. On adversarial environments, SSAC suppresses hidden-objective hacking that GDPO and GRPO-sum learn to execute. SSAC is a drop-in replacement for the advantage construction in any GRPO-based RLVR algorithm.

1 Introduction

Policy gradient-based Reinforcement Learning in LLMs has resulted in improvements in a large variety of tasks. Recent works have used multi-objective RL to align Large Language Models for preferred behaviors like instruction following and alignment. Auxiliary process rewards have also shown to improve performance in mathematical reasoning tasks (Lightman et al., 2023). These have shown to be effective particularly in verifiable regimes with binary outcomes.

Reinforcement learning with verifiable rewards (RLVR) has emerged as the dominant post-training paradigm for language model reasoning, with deterministic outcome verifiers — answer-matching for math, test execution for code — providing the primary training signal (Lambert et al., 2024; DeepSeek-AI et al., 2025). Within Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and its variants (Yu et al., 2025; Liu et al., 2025), the binary verifier produces precise but sparse feedback: when all rollouts in a group share the same correctness, the within-group standardization vanishes and no gradient signal is produced. Practitioners address this by composing the verifier outcome with auxiliary signals — process reward models that score reasoning steps (Lightman et al., 2023; Zhang et al., 2025), format and length penalties, and rubric-derived scalars — that provide dense within-group differentiation even where the verifier alone does not. Multi-reward composition is now standard practice. GDPO (Liu et al., 2026) combines binary outcome and PRM z-scores additively across tool-calling, math, and code; PRIME (Cui et al., 2025) maintains an online-updated implicit reward additively composed with the verifier; ToolRL (Qian et al., 2025) combines correctness and format signals for tool-use; RaR (Gunjal et al., 2025) aggregates itemized rubric criteria into a single scalar. The number of signals varies from two to dozens, but the structural question is the same in every case: how should an auxiliary signal modify the per-rollout advantage produced by a trusted verifier?

Sign-disagreeing advantages. We show that additive reward aggregation has a structural pathology. The outcome term u_i in the GRPO advantage agrees in sign with the verifier by construction, whereas the auxiliary term λv_i does not. When $|\lambda v_i| > |u_i|$ the composed advantage $A_i = u_i + \lambda v_i$ can carry a sign opposite to the verifier direction $2y_i - 1$. We call such rollouts *sign-disagreeing*. Figure 1(a) illustrates the mechanism: a correct rollout with a low auxiliary score receives a negative advantage, while an incorrect

(a) Additive auxiliary rewards can reverse the verifier sign

Example with $\lambda = 1.0$ and $\beta = 0.5$: B is correct but low-PRM; C is incorrect but high-PRM.

	Verifier u_i	Auxiliary v_i	GDPO $u_i + \lambda v_i$	SSAC aux $\lvert\tau_{y_i} v_i\rvert \leq \beta \lvert u_i \rvert$
B: correct low PRM	$u_B = +1$	$v_B = -1.21$	$A_B = -0.21$ **pushes down**	$A_B = +0.50$ preserved
C: incorrect high PRM	$u_C = -1$	$v_C = +1.21$	$A_C = +0.21$ **pushes up**	$A_C = -0.50$ preserved

Takeaway: additive composition can reverse the trusted verifier direction; SSAC changes magnitude without changing the verifier-implied sign.

(b) Empirical sign-disagreement rate

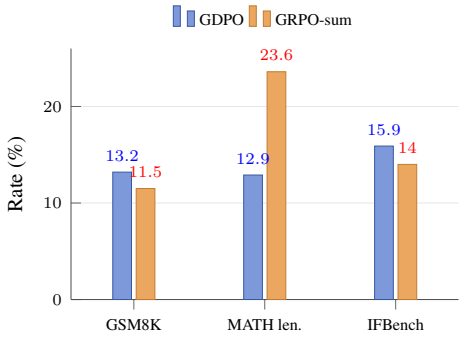

Figure 1: **Mechanism and prevalence of sign disagreement. (a)** A simplified worked example showing how additive reward composition can give a negative advantage to a correct rollout and a positive advantage to an incorrect rollout. SSAC preserves the trusted verifier sign by bounding the auxiliary contribution. **(b)** Empirical sign-disagreement rates for GSM8K/Qwen2.5-Math-1.5B, MATH-length/Qwen3-4B, and Real IFBench/Qwen3-4B.

rollout with a high auxiliary score receives a positive one.

We measure this empirically across three tasks. Additive composition produces sign-disagreeing advantages on approximately 8–39% of rollouts; see Figure 1(b) and the full audit in Table 1. The rates are 38.9% on MATH with PRM800K, 30.2% on MATH with PRM-7B, and 8.6% on IFBench. The rate scales with the auxiliary weight λ in GDPO and depends on the auxiliary reward's per-rollout correlation with the verifier. Figure 1 demonstrates the mechanism of sign flipping through an example.

Spurious auxiliary signals dominating verified outcomes is a recognized failure mode in reward-based LLM training, previously addressed at the reward model layer in RLHF and at the data filtering layer in RLVR (§2). SSAC operates at the advantage construction layer without reward model retraining or rollout discarding. This positioning is what enables a single-parameter drop-in replacement for the additive composition rule, rather than a separate training stage or data pipeline modification.

Each sign-disagreeing rollout contributes a policy gradient term $A_i \nabla_\theta \log \pi_\theta(\hat{y}_i \mid x)$ with A_i opposite in sign to the verifier's judgment — a gradient contribution pointing toward a verifier-rejected output. Whether the aggregate update direction is corrupted depends on the magnitude and distribution of these terms across the batch, which we do not claim without measurement. What we do claim is structural: additive composition has no mechanism to prevent individual gradient terms from carrying the wrong sign, and at standard hyperparameters this affects a substantial fraction of every batch.

Empirically, this structural property is associated with unstable training. In our MATH experiments training on Qwen3-4B-2507 with PRM scores as auxiliary rewards, GDPO collapses catastrophically to zero outcome reward. Naively composed GRPO also degrades from the baseline. We propose SSAC, an advantage construction that eliminates sign-disagreeing advantages by design. SSAC modulates the auxiliary signal within each outcome class but structurally bounds its per-rollout magnitude below the outcome anchor magnitude. The construction enables a family of algorithms and future directions, and we evaluate the simplest constructions of this family which empirically outperform currently used ways of reward aggregation. We elaborate on this family in the A but defer experimentation over this to future work.

SSAC addresses the foundational case of one binary verifier and one continuous auxiliary. Extension to multiple auxiliary signals requires either pre-aggregation of auxiliaries before applying SSAC, or a multi-auxiliary generalisation with budget allocation per signal; we defer this to future work

2 Related Work

Multi-reward composition in RLVR. Two composition rules dominate: decoupled normalization (GDPO; Liu et al., 2026), which sums per-reward z-scores, and joint normalization (GRPO-sum), which z-scores the weighted sum (variants in DAPO (Yu et al., 2025) and PRIME (Cui et al., 2025)). Both are baselines in our experiments and

both produce sign-disagreeing advantages (§5.2). PROF (Ye et al., 2025) addresses the same failure mode at the data layer by filtering rollouts where the verifier and PRM disagree; SSAC retains all rollouts and bounds the auxiliary's per-rollout contribution instead. PRIME additionally maintains an online-updated implicit PRM to mitigate staleness, but its composition with the verifier remains additive. PRPO (Ding et al., 2026) aligns process and outcome distributions at the token level, addressing a distinct mismatch; ALARM (Lai et al., 2024) composes preference and aspect rewards via threshold-based hierarchy in RLHF.

Failure mode in adjacent settings ODIN (Chen et al., 2024) prevents length-quality contamination in RLHF by training a two-headed reward model with an orthogonality loss; SSAC operates at the advantage layer instead, requiring no reward model retraining and providing a per-rollout structural guarantee rather than statistical decorrelation. PAR (Fu et al., 2025) bounds proxy rewards before they enter RLHF training; SSAC's per-rollout bound is class-conditional and applied at advantage construction.

Reward hacking and PRM overoptimization. Gao et al. (2023) establish scaling laws for proxy reward overoptimization in RLHF. Tiwari et al. (2026) measure that PRMs in RLVR function partly as fluency detectors, with 43% of PRM-driven reward gains attributable to stylistic shortcuts. These results motivate the structural caution SSAC imposes: the verifier should retain sign authority over the advantage even when the auxiliary disagrees, particularly when the auxiliary is a learned PRM susceptible to overoptimization.

Advantage construction in RL for LLMs. GRPO (Shao et al., 2024) introduced within-group standardization as the advantage construction in RLVR; variants include Dr.GRPO (Liu et al., 2025) (removing standard-deviation normalization), DAPO (Yu et al., 2025) (clip-higher, dynamic sampling), RLOO (Ahmadian et al., 2024), and REINFORCE++ (Hu et al., 2025). SSAC is a drop-in replacement for the advantage construction in any of these.

3 Preliminary

3.1 Group-relative policy optimization

Reinforcement learning with verifiable rewards (RLVR) optimizes a language model policy π_θ on prompts x drawn from a distribution $\mathcal{D}$. For each prompt, a group of N rollouts $\{\hat{y}_i\}_{i=1}^N$ is sampled from $\pi_\theta(\cdot \mid x)$ and each receives a binary outcome reward $y_i \in \{0, 1\}$ from a deterministic verifier — answer-matching for math, test execution for code.

Group Relative Policy Optimization (Shao et al., 2024) computes a within-group standardized advantage,

$$A_i^{\mathrm{GRPO}} = \frac{y_i - \bar{y}}{\sigma_y}, \tag{1}$$

where $\bar{y} = \frac{1}{N}\sum_i y_i$ and σ_y is the within-group standard deviation. The policy is updated by maximizing the clipped surrogate objective with A_i as the per-rollout weight on $\nabla_\theta \log \pi_\theta(\hat{y}_i \mid x)$.

For binary outcomes with $k = \sum_i y_i$ correct rollouts and $m = N - k$ incorrect ones, (1) reduces to a class-constant anchor:

$$u_i = \begin{cases} +\sqrt{m/k}, & y_i = 1, \\ -\sqrt{k/m}, & y_i = 0, \end{cases} \tag{2}$$

satisfying $\frac{1}{N}\sum_i u_i = 0$ and $\frac{1}{N}\sum_i u_i^2 = 1$. The sign of u_i is the verifier's update direction: $\mathrm{sign}(u_i) = 2y_i - 1$. When $k \in \{0, N\}$, the group is *degenerate*: all rollouts share the same outcome, $\sigma_y = 0$, and (1) is undefined; in this case GRPO assigns $A_i = 0$ to every rollout and no policy update occurs. Variants of GRPO — Dr.GRPO (Liu et al., 2025), DAPO (Yu et al., 2025), RLOO (Ahmadian et al., 2024), REINFORCE++ (Hu et al., 2025) — modify other aspects of training (clipping, sampling, normalization) but retain the same advantage construction.

3.2 Multi-reward composition

In multi-reward RLVR, the binary verifier outcome y_i is supplemented by an auxiliary reward $b_i \in R$: a process reward model (PRM) score, a format or length signal, or a rubric-derived scalar. The auxiliary provides dense within-group differentiation even in degenerate groups where the verifier alone produces no gradient signal. Two composition rules dominate the literature.

Decoupled normalization (GDPO). Liu et al. (2026) normalize each reward independently within the group, then sum the results with a weight λ:

$$A_i^{\mathrm{GDPO}} = \frac{y_i - \bar{y}}{\sigma_y} + \lambda \frac{b_i - \bar{b}}{\sigma_b}. \tag{3}$$

The auxiliary z-score $(b_i - \bar{b})/\sigma_b$ is normalized to unit variance regardless of the auxiliary's native

scale, allowing the auxiliary term to dominate the outcome term when λ is large.

Joint normalization (GRPO-sum). The weighted sum is formed first, then standardized as a single signal:

$$A_i^{\text{GRPO-sum}} = \frac{(y_i + \lambda b_i) - \overline{y + \lambda b}}{\sigma_{y+\lambda b}}. \quad (4)$$

Variants of this rule appear in DAPO (Yu et al., 2025) and as the baseline formulation in PRIME (Cui et al., 2025); the joint denominator $\sigma_{y+\lambda b}$ couples the two signals' variance contributions.

The two rules are structurally distinct. (3) preserves each reward's resolution by normalizing them separately; (4) preserves the combined signal's variance structure but loses per-reward resolution. Both are widely used in deployed multi-reward RLVR pipelines, and both produce the sign-disagreement failure mode we characterize in §**??**. We evaluate both as baselines throughout the paper.

3.3 Setting and notation

We consider a single continuous auxiliary $b_i \in R$ per rollout paired with the binary verifier outcome $y_i \in \{0, 1\}$. We write $\mathcal{I}_c = \{i : y_i = c\}$ for the index set of rollouts with outcome c, with sizes $k = |\mathcal{I}_1|$ and $m = |\mathcal{I}_0|$. A group is *degenerate* if $k \in \{0, N\}$ and *non-degenerate* otherwise. An advantage rule $\mathcal{A} : \{(y_i, b_i)\}_{i=1}^N \mapsto \{a_i\}_{i=1}^N$ produces the per-rollout advantages entering the policy gradient. Two questions structure the rest of the paper: (a) whether $\text{sign}(a_i)$ agrees with the verifier $2y_i - 1$ for every rollout (§5.2), and (b) how to construct an advantage rule that guarantees this agreement while still allowing the auxiliary to modulate update magnitude (§4). Extension to multiple auxiliary signals requires either pre-aggregation of auxiliaries before applying the rule or a multi-auxiliary generalization with per-signal budget allocation; we restrict attention to the single-auxiliary case throughout.

4 Sign-Safe Advantage Composition

We now construct SSAC, an advantage composition rule that extracts within-class information from the auxiliary signal while structurally preventing it from reversing the verifier's update direction on any rollout.

Setup. For each prompt, let $\{(y_i, b_i)\}_{i=1}^N$ denote a group of N rollouts, where $y_i \in \{0, 1\}$ is the trusted outcome reward from a deterministic verifier and $b_i \in R$ is a continuous auxiliary reward. Write $k = \sum_i y_i$, $m = N - k$, and $\mathcal{I}_c = \{i : y_i = c\}$ for $c \in \{0, 1\}$. A group is *degenerate* when $k \in \{0, N\}$: every rollout shares the same outcome and the verifier provides no within-group contrast. In this case SSAC returns $a_i = 0$ for all i, matching GRPO's behavior. The remainder of this section describes SSAC for non-degenerate groups.

Outcome anchor. SSAC first constructs the binary outcome anchor

$$u_i = \begin{cases} +\sqrt{m/k}, & y_i = 1, \\ -\sqrt{k/m}, & y_i = 0. \end{cases} \quad (5)$$

This is the outcome-only GRPO advantage. It is class-constant, zero-mean, and unit-RMS: $\frac{1}{N}\sum_i u_i = 0$ and $\frac{1}{N}\sum_i u_i^2 = 1$. Its sign encodes the verifier's update direction, $\text{sign}(u_i) = 2y_i - 1$, and its magnitude $\gamma_c \equiv |u_i|$ for $i \in \mathcal{I}_c$ depends only on the class size: $\gamma_1 = \sqrt{m/k}$ and $\gamma_0 = \sqrt{k/m}$.

Within-class auxiliary direction. The auxiliary signal modulates magnitude within each outcome class. SSAC first standardizes b_i within its class. For $c \in \{0, 1\}$ with $|\mathcal{I}_c| \geq 2$, let $\mu_c = \frac{1}{|\mathcal{I}_c|}\sum_{j \in \mathcal{I}_c} b_j$ and $\sigma_c = \sqrt{\frac{1}{|\mathcal{I}_c|}\sum_{j \in \mathcal{I}_c}(b_j - \mu_c)^2}$. The within-class score is

$$\phi_i = \begin{cases} \dfrac{b_i - \mu_{y_i}}{\sigma_{y_i} + \epsilon}, & |\mathcal{I}_{y_i}| \geq 2,\ \sigma_{y_i} > 0, \\ 0, & \text{otherwise}, \end{cases} \quad (6)$$

with ϵ a small numerical stabilizer. Within-class standardization gives $\sum_{i \in \mathcal{I}_c} \phi_i = 0$ for non-degenerate classes, so the auxiliary redistributes advantage among correct rollouts and among incorrect rollouts separately, without shifting either class mean. The construction is also invariant to within-class affine recalibration of the auxiliary: replacing b_i by $\alpha_c b_i + \delta_c$ for any $\alpha_c > 0$ leaves ϕ_i unchanged. SSAC then RMS-normalizes ϕ across the full group to obtain the auxiliary direction

$$v_i = \frac{\phi_i}{\sqrt{\frac{1}{N}\sum_j \phi_j^2} + \epsilon}. \quad (7)$$

Sign-safe scaling. The auxiliary direction v may change the magnitude of the update, but not its sign. SSAC enforces this with a per-class scaling

factor that bounds the auxiliary contribution strictly below the outcome anchor magnitude. Let $B_c = \max_{i \in \mathcal{I}_c} |v_i|$ denote the largest auxiliary direction in class c, and let $\beta \in (0, 1)$ be a safety margin. SSAC sets

$$\tau_c = \beta \cdot \frac{\gamma_c}{B_c} \quad \text{when } B_c > 0, \tag{8}$$

and $\tau_c = 0$ otherwise. The raw composed advantage is

$$\tilde{a}_i = u_i + \tau_{y_i} v_i, \tag{9}$$

and the final SSAC advantage is its group RMS normalization:

$$a_i = \frac{\tilde{a}_i}{\sqrt{\frac{1}{N} \sum_j \tilde{a}_j^2} + \epsilon}. \tag{10}$$

Sign safety guarantee. The choice of τ_c ensures that the auxiliary contribution is strictly smaller than the outcome anchor magnitude for every rollout. For any i in a non-degenerate class:

$$|\tau_{y_i} v_i| \leq \beta \cdot \frac{\gamma_{y_i}}{B_{y_i}} \cdot B_{y_i} = \beta |u_i| < |u_i|. \tag{11}$$

Therefore $\tilde{a}_i$ and u_i share their sign, and since RMS normalization multiplies all advantages by a positive scalar,

$$\text{sign}(a_i) = \text{sign}(u_i) = 2y_i - 1 \tag{12}$$

for every rollout. SSAC has zero sign-disagreement rate by construction on every non-degenerate group, regardless of the auxiliary signal's calibration, range, or correlation with the outcome.

Reduction to GRPO. When the auxiliary is constant within each outcome class — including the limiting case of a vanishing auxiliary — the within-class standardization gives $\phi = 0$, hence $v = 0$ and $\tilde{a} = u$. Since u is already zero-mean and unit-RMS, the final normalization is the identity up to the stabilizer, and SSAC reduces exactly to outcome-only GRPO. SSAC therefore inherits GRPO's behavior as a strict special case, with the auxiliary signal contributing nothing when it carries no within-class information.

Interpretation. The verifier outcome and the auxiliary signal play asymmetric roles in SSAC. The verifier determines, through u_i, the sign of every rollout's update — whether the policy is pushed toward or away from the response. The auxiliary determines, through $\tau_{y_i} v_i$, a bounded within-class redistribution of magnitude: among correct rollouts, those with higher auxiliary scores receive larger positive advantages; among incorrect rollouts, those with lower auxiliary scores receive more negative advantages. The auxiliary cannot turn an incorrect rollout into a positive update or a correct rollout into a negative one. The safety margin β controls how much within-class information the auxiliary contributes: $\beta \to 0$ recovers GRPO; $\beta \to 1$ approaches the sign-safety boundary; intermediate values trade off auxiliary utilization against margin to the boundary.

5 Experiments

We evaluate SSAC against three baselines — outcome-only GRPO, decoupled additive composition (GDPO), and joint additive composition (GRPO-sum) — across five experimental regimes ordered by auxiliary signal reliability: a structural offline audit (§5.2); training collapse under adversarial PRM-driven hacking on MATH (§5.3); SSAC's outperformance on the hardest problem tier where the auxiliary is informative (§5.4); designed adversarial reward-hacking environments (§5.5); and aligned-auxiliary regimes where SSAC's bound is not load-bearing (§5.6). §5.7 provides a β sensitivity analysis.

5.1 Setup

Models and tasks. We use Qwen3-4B-Instruct-2507 for MATH and IFBench experiments and Llama-3.2-1B-Instruct for an adversarial reward-hacking suite. MATH experiments use the L3–L5 split for training and a balanced L1–L5 evaluation. Auxiliary signals span (i) Qwen2.5-Math-7B-PRM800K process reward model, (ii) partial constraint satisfaction on IFBench, and (iii) hidden-objective rewards in environments where hacking is encouraged.

Training protocol. Group size $N = 16$, 4 groups per batch, max completion 2048 tokens, LoRA rank 32, learning rate 8×10^{-5}, KL coefficient 5×10^{-4} unless noted. For GDPO we sweep $\lambda \in \{0.05, 0.25\}$ which are on the safer side of the recommended range of values; we report both. For GRPO-sum we use $\lambda = 1.0$. For SSAC we use $\beta = 0.5$ as a default, with sensitivity analysis in §5.7. All experiments train for 200 steps.

Task / Auxiliary	GRPO	GRPO-sum	GDPO	SSAC
MATH + PRM800K	0.000	0.000	0.389	**0.000**
MATH + PRM-7B	0.000	0.244	0.302	**0.000**
IFBench	0.000	0.083	0.086	**0.000**
Misdirected mass				
MATH + PRM800K	0.000	0.000	0.353	**0.000**
MATH + PRM-7B	0.000	0.280	0.311	**0.000**

Table 1: **Offline sign-disagreement audit.** Top block: fraction of rollouts where the advantage sign disagrees with the verifier outcome. Bottom block: fraction of total absolute advantage magnitude assigned in the verifier-rejected direction. SSAC eliminates both by construction; additive composition produces non-trivial rates across all three task-auxiliary pairings.

Method	Final	PRM	Len	Flip	Hacked?
GRPO	0.984	0.646	447	0.00	—
SSAC (β=0.5)	**0.984**	0.611	459	0.00	—
GRPO-sum (λ=1)	0.781	0.684	828	0.65	partial
GDPO (λ=0.05)	**0.000**	0.760	2048	1.00	**yes**
GDPO (λ=0.25)	**0.000**	0.758	2047	1.00	**yes**

Table 2: **MATH + PRM800K training, 200 steps.** Final outcome accuracy, PRM mean, mean completion length (words), batch-level sign-flip rate, and operational hacking label. Both published GDPO λ values collapse outcome to zero while maximizing PRM and length. SSAC matches GRPO outcome-only while the auxiliary remains available for within-class modulation.

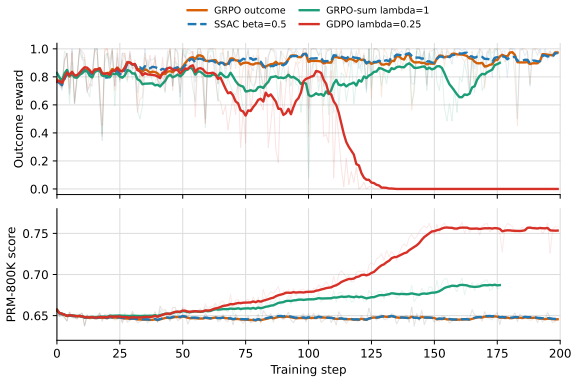

Figure 2: **MATH + PRM800K outcome accuracy over training.** SSAC ($\beta = 0.5$) and GRPO-outcome converge cleanly, GDPO collapses early

Method	Acc.	L1	L2	L3	L4	L5
GRPO	0.725	0.950	0.800	0.800	0.700	0.375
SSAC	**0.755**	0.975	0.775	0.850	0.675	**0.500**
GRPO-sum	0.500	0.825	0.600	0.575	0.350	0.150
GDPO	0.145	0.375	0.200	0.100	0.050	0.000

Table 3: **Balanced MATH L1–L5 evaluation.** Accuracy by difficulty tier, all methods trained on the same MATH L3–L5 split with the PRM800K auxiliary. SSAC outperforms outcome-only GRPO by +3.0 pp overall and +12.5 pp on the hardest tier (L5). GDPO and GRPO-sum underperform on every tier.

5.2 Sign Disagreement is Widespread

We first measure how frequently each composition rule produces an advantage with sign opposite to the verifier outcome. The audit is offline: it scores cached rollouts under each rule without training a policy. Across four diverse task-auxiliary pairings, additive composition produces sign-disagreeing advantages on 7–39% of rollouts; SSAC produces zero (Table 1).

We claim two things from this audit. First, sign disagreement is not an artifact of a specific PRM or task: it appears across two different PRMs and instruction-following reward configurations. Second, the rate is high enough on math PRMs (24–39%) that a non-trivial fraction of every training batch contains misdirected gradient contributions.

5.3 GDPO Collapses Training Under PRM Auxiliaries; SSAC Does Not

We train on MATH L3–L5 with the Qwen2.5-Math-7B-PRM800K process reward as auxiliary. Both GDPO λ values from the original paper produce catastrophic training collapse (Table 2).

Both GDPO runs reach 0.000 outcome accuracy by step 200 while raising PRM by +0.11 over GRPO and inflating mean response length to the maximum generation budget. Wrong-positive advantage rates reach 1.00 at the batch level. The policy learned to maximize PRM at the expense of correctness, the canonical reward-hacking failure mode, at the λ values published in the GDPO paper for math-style tasks. SSAC at $\beta = 0.5$ holds outcome at 0.984 with normal length.

Figure 2 shows the training trajectories. GDPO's outcome reward falls monotonically while PRM rises; SSAC's outcome and PRM track GRPO throughout.

5.4 SSAC Outperforms on the Hardest Problems

When evaluated across MATH difficulty tiers L1–L5, the SSAC vs. GRPO comparison is most informative on the hardest tier, where the auxiliary PRM signal contributes the most within-class information. Table 3 shows the breakdown.

The L5 gap (+12.5 pp) is the clearest evidence that the auxiliary signal carries useful within-class

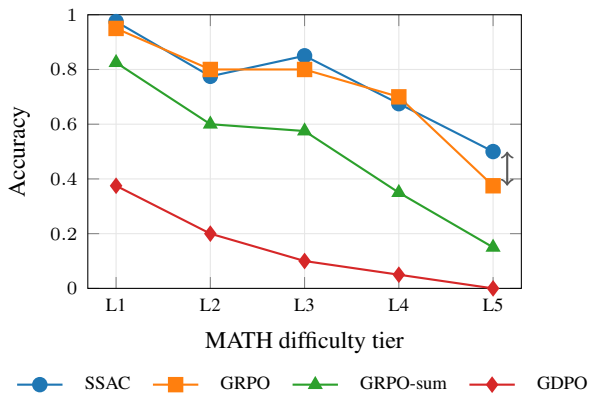

Figure 3: **MATH accuracy by difficulty tier.** SSAC matches GRPO outcome-only on easy tiers (L1–L4) and outperforms it by +12.5 pp on L5, where the auxiliary signal carries the most within-class information. Additive composition (GDPO, GRPO-sum) degrades sharply with difficulty, with GDPO reaching 0% accuracy on L5.

Env.	Method	Hidden	Visible	Hacked?
agg-restrict	GDPO	**0.863**	0.175	**yes**
agg-restrict	SSAC (β=0.1)	**0.000**	0.381	**no**

Table 4: **Llama-3.2-1B reward-hacking environments.** Hidden reward measures backdoor-objective optimization (lower is safer); visible reward measures instruction-following on the benign task. GDPO on agg-restrict execute the hack (hidden reward ≥ 0.86). SSAC suppresses the hack entirely (hidden reward 0.000) while maintaining nonzero visible reward. A detailed explanation into the environment and reward formulation set up can be found in Appendix D.

information when it is bounded rather than allowed to dominate. On easier tiers (L1–L4) the gap is smaller or slightly negative for SSAC, reflecting that the auxiliary's contribution is most valuable when the verifier outcome is sparsest — which is exactly the hard-problem regime.

Figure 3 visualizes the per-tier comparison.

5.5 Adversarial Reward Hacking

We evaluate on adversarial reward-hacking environments on Prime Lab (Li, 2026) designed to elicit hidden-objective optimization. Each environment provides a visible reward (instruction-following on a benign task) and a hidden reward (a backdoor objective the model can learn to optimize through specific patterns). A method is hacked when hidden reward rises while visible reward holds or falls.

The Prime Lab demonstration is narrow but specific, additive composition learns the hidden objective while SSAC does not hack. The hidden reward gap between GDPO (0.863) and SSAC (0.000) on the same environment is the cleanest available isolation of the sign-safety mechanism's effect in an adversarial setting. The expanded suite (currently running) will report this comparison across the full environment-method grid.

Setting / Method	Outcome	Aux	Flip	Len
MATH PRM held-out (40 groups, final checkpoint)				
GRPO	0.861	0.649	0.00	531
SSAC	**0.873**	0.650	0.00	541
IFBench mixed constraints				
GRPO	0.750	—	0.00	256
GRPO-sum	0.938	0.969	0.00	226
GDPO	0.938	0.969	0.00	239
SSAC	**1.000**	1.000	0.00	242

Table 5: **Aligned-auxiliary regimes.** SSAC demonstrates better training dynamics in misaligned auxiliary settings; to demonstrate SSAC's robustness we run experiments in regimes where the auxiliary reward is well-aligned to evaluate SSAC's performance in ideal settings. On MATH PRM held-out, SSAC modestly outperforms GRPO (+1.25 pp). On IFBench mixed (single seed), SSAC reaches 1.000 outcome where additive composition methods reach 0.938

5.6 Aligned-Auxiliary Regimes: SSAC Does Not Damage Training

We evaluate on two settings where the auxiliary signal is well-aligned with the verifier and additive composition does not produce sign disagreement at training time: held-out MATH PRM evaluation and IFBench with partial-constraint auxiliary. Most importantly, naive GRPO-sum also outperforms outcome-only GRPO, which demonstrates that the auxiliary signal improves the primary reward accuracy.

On MATH held-out, SSAC modestly outperforms GRPO with identical PRM and length. On IFBench, all multi-reward methods improve over outcome-only GRPO, with SSAC slightly ahead. We do not over-interpret the IFBench single-seed advantage; the key observation is that SSAC's bound does not prevent the method from extracting available auxiliary value when no sign disagreement occurs.

5.7 Sensitivity to β

The safety margin β controls how much within-class auxiliary information SSAC extracts: $\beta \to 0$ recovers GRPO outcome-only; $\beta \to 1$ approaches the sign-safety boundary. We examine hyperparameter sensitivity for Prime Lab's adver-

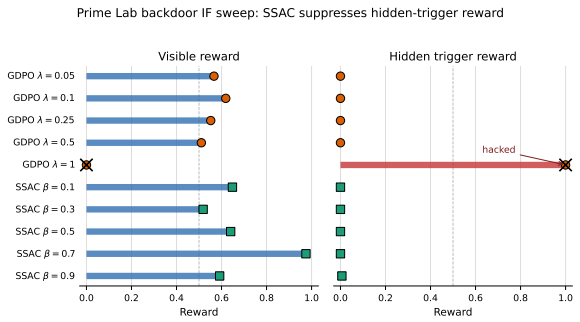

Figure 4: We train Llama-3.2-1B-Instruct on the hosted IF environment with the frequency of the word "silver" being a hidden objective. We find that GDPO collapsed at $\lambda = 1$ and SSAC outperformed GDPO by +11.3 pp after discounting the hacking case, and +3.8 pp after discounting the outlier $\beta = 0.7$ SSAC run

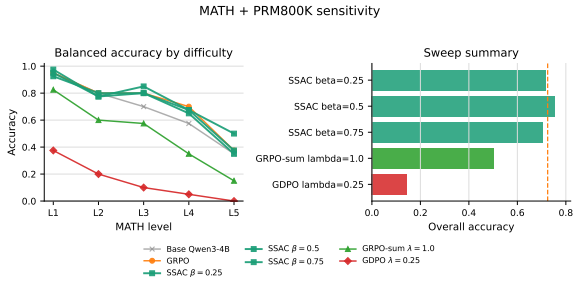

Figure 5: We rerun the MATH experiment in Sections 5.3 and 5.4. Additive composition over-optimizes the PRM: GRPO-sum/GDPO raise PRM to $0.691/0.705$ and length to $648/826$, but accuracy falls to $0.500/0.145$. SSAC avoids this tradeoff. Its sweep matches GRPO on average (0.725 acc.) and $\beta = 0.5$ is the best setting, improving accuracy to 0.755 and L5 to 0.500 without increasing PRM or length.

sarial instruction-following environments. Figure 4 shows adversarial IF accuracy across $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and for GDPO's $\lambda \in \{0.05, 0.1, 0.25, 0.5, 1.0\}$.

From Figure 5 we see that SSAC's performance on the held-out set is consistent and improving on the base model and outcome-only GRPO. Naive composition and GDPO reduce performance. A limitation we highlight here is that GDPO for MATH has not been swept for hyperparameters, but the consistency in SSAC's result gives us a strong directional signal.

The β sweep demonstrates that SSAC's performance is not knife-edge, but a carefully chosen β will find performance gains

6 Conclusion

This paper studies a failure mode and a design principle for multi-objective RLVR. Specifically, we target settings where one verifier is deterministic and trusted, and others are approximate such as PRMs, rubrics, formats, and length penalties. In these settings, auxiliary signals should not be allowed to dominate the policy-update direction and the trusted signal should retain sign authority. We argue that auxiliary rewards modulating the magnitude of the updates is a more effective technique and a more principled manner of studying auxiliary rewards. Additive composition, the dominant rule in the current literature, does not satisfy this principle, and the resulting sign-disagreement is frequent enough at standard hyperparameters to drive catastrophic training collapse under common process reward models. SSAC operationalizes this principle as a single-parameter advantage rule. Its broader implication is that the apparent tension between dense auxiliary feedback and verifier-faithful learning is not fundamental: a structural per-rollout bound resolves it without discarding rollouts, retraining reward models, or modifying the policy gradient framework. This work is a step towards understanding how tens of rubric criteria should be aggregated for optimal training. Through this paper we show that advantage construction is an underutilized component in RLVR objective formulation and encourage future work to focus on advantage construction which encompasses multiple reward signals.

7 Limitations

Our work has limitations that are important to contextualize.

Single-auxiliary setting. SSAC as formulated addresses the case of one binary verifier paired with one continuous auxiliary signal. Multi-auxiliary settings such as aggregated rubric criteria (Gunjal et al., 2025; Qian et al., 2025) require either pre-aggregation of auxiliaries before applying SSAC or a generalization with per-signal budget allocation.

Seed coverage. Results in several tables are reported on 1-3 seeds. The magnitude of empirical improvements (notably the $+12.5$ pp gain on MATH Level 5 and the IFBench results) should be interpreted as indicative rather than tight estimates of expected gains.

Ethical Considerations

This work studies the advantage construction in reinforcement learning with verifiable rewards. Experiments use publicly released models and benchmarks (MATH, IFBench, and the Prime Lab reward-hacking suite) and involve no human subjects, no data collection, and no release of new model artifacts. The contribution is a structural property intended to reduce reward hacking in multi-reward RLVR, which we view as broadly aligned with the safe deployment of language model post-training.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ODIN: Disentangled reward mitigates hacking in RLHF. In *Proceedings of the 41st International Conference on Machine Learning*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ruiyi Ding, Yongxuan Lv, Xianhui Meng, Jiahe Song, Chao Wang, Chen Jiang, and Yuan Cheng. 2026. PRPO: Aligning process reward with outcome reward in policy optimization. *arXiv preprint arXiv:2601.07182*.

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

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.

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

Wenbo Hu, Xin Chen, Yan Gao-Tian, Yihe Deng, Nanyun Peng, and Kai-Wei Chang. 2026. Open-vlthinkerv2: A generalist multimodal reasoning model for multi-domain visual tasks. *arXiv preprint arXiv:2604.08539*.

Yuhang Lai, Siyuan Wang, Shujun Liu, Xuanjing Huang, and Zhongyu Wei. 2024. ALaRM: Align language models via hierarchical rewards modeling. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Jessica Li. 2026. Systematic reward hacking and prime sprints. `https://www.primeintellect.ai/blog/reward-hacking`.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. 2026. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. ToolRL: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Rishabh Tiwari, Aditya Tomar, Udbhav Bamba, Monishwaran Maheswaran, Heng Yang, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2026. Reward under attack: Analyzing the robustness and hackability of process reward models. *arXiv preprint arXiv:2603.06621*.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. 2025. Beyond correctness: Harmonizing process and outcome rewards through RL training. *arXiv preprint arXiv:2509.03403*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

A SSAC Family

The sign-safety property derived in §4 depends only on the per-class bound $\tau_c = \beta\gamma_c/B_c$, where B_c is the maximum within-class magnitude of the auxiliary direction. The construction of ϕ_i — the within-class transformation of b_i that yields that direction — is otherwise unconstrained. SSAC is therefore best understood as a family of advantage rules parameterised by the choice of modulation $\phi : R \to R$, applied within each outcome class. The variant evaluated throughout this paper uses within-class z-scoring; we describe several alternatives that satisfy the same sign-safety guarantee.

Within-class z-scoring (this paper). $\phi_i = (b_i - \mu_{y_i})/\sigma_{y_i}$. The auxiliary signal is centered and rescaled to unit variance within each class. This is the variant we evaluate.

Dr.GRPO-style centering. $\phi_i = b_i - \mu_{y_i}$. Centering without variance normalisation, analogous to Liu et al. (2025)'s treatment of the outcome term. Useful when the auxiliary's native scale is meaningful and the variance normaliser is noisy in small groups.

Quantile transformation. Map b_i to its within-class rank $r_i \in \{1, \ldots, |\mathcal{I}_{y_i}|\}$ and let $\phi_i = \Phi^{-1}(r_i/(|\mathcal{I}_{y_i}| + 1))$, where Φ^{-1} is the standard normal quantile function. This is the within-group rank transform used by G^2RPO (Hu et al., 2026) and similar approaches; it is robust to outliers in the auxiliary distribution.

Exponential tilt. $\phi_i = \exp(\eta b_i)/Z_{y_i} - 1/|\mathcal{I}_{y_i}|$, where $Z_{y_i} = \sum_{j \in \mathcal{I}_{y_i}} \exp(\eta b_j)$. The temperature η controls how sharply auxiliary weight concentrates on the highest-scoring within-class rollouts; the subtraction recenters ϕ within the class.

Mean-shifted variants. Any modulation above can be combined with a positive shift $\phi_i \mapsto \phi_i + c$ to introduce a class-conditional positive bias before the sign-safe bound is applied. This is useful when the auxiliary should systematically reinforce same-class rollouts before being clipped to the safety margin.

General property. For any choice of ϕ, after RMS-normalisation to produce v_i and the per-class scaling τ_c, the sign-safety guarantee $|\tau_{y_i} v_i| < |u_i|$ holds by construction (§4). The choice of ϕ controls only how the auxiliary distributes magnitude among same-class rollouts; it cannot affect verifier sign authority. The variants above are reasonable starting points within this family, of which we evaluate only z-scoring; broader experimentation is deferred to future work.

B Instruction-Following Experimental Configuration

Data curation. We first ran a local vLLM audit on IFBench using Qwen3-4B-Instruct-2507 with $N = 16$ rollouts per prompt, minimum two constraints per prompt, temperature 0.9, top-$p = 0.95$, and maximum completion length 256. The audit covered 400 prompts and 6400 rollouts. The resulting pool had all-pass rate 0.3009, mean partial-pass score 0.5677, mixed-group rate 0.2025, and mean $k = 4.815$ successful rollouts out of 16. From this audit we randomly sampled the first 256 selected prompts.

Quantity	Value
Dataset	`sungyub/ifbench-verl`
Split	train
Base model	Qwen3-4B-Instruct-2507
Audit prompts	400
Audit rollouts per prompt	16
Training prompts	256
Minimum constraints	2
Training selection seed	31
Audit all-pass rate	0.3009
Audit partial-pass mean	0.5677
Audit mixed-group rate	0.2025

Table 6: IFBench audit and training-selection statistics.

Split	Examples	Levels	Construction
Train	165	L3–L5	Mixed-outcome groups
Holdout	40	L3–L5	Deterministic every-nth row
Eval	200	L1–L5	40 problems per level

Table 7: MATH data splits used for training and evaluation.

C MATH Experimental Configuration

Task and base model. All MATH training experiments use Qwen3-4B-Instruct-2507 as the policy model. The task is boxed-answer mathematical reasoning: for each problem, the model is prompted to reason step by step and to place the final answer inside `\boxed{}`. The trusted outcome reward is exact final-answer correctness after boxed-answer extraction and answer normalization. The auxiliary reward is the trajectory-level process reward from Qwen2.5-Math-7B-PRM800K. The PRM is applied locally to each completed response: the response is split into reasoning steps, the PRM assigns a score to each step marker, and the auxiliary reward used for training is the mean step score.

Training-data curation. We first ran an offline task-discovery pass with Qwen3-4B-Instruct-2507 on the competition MATH training split. The discovery pass generated $N = 16$ rollouts per problem at temperature 0.8, top-$p = 0.95$, and a maximum of 1024 completion tokens. We retained only groups with mixed binary outcomes, i.e. $1 \leq k \leq 15$ correct rollouts out of 16, because such groups provide a nonzero outcome-learning signal and make advantage sign disagreements observable. From the resulting mixed-group pool, we filtered to MATH Levels 3–5 and sorted examples by $|k - 8|$, prioritizing groups close to a balanced success rate. We then assigned a deterministic holdout by taking every $\lfloor n/40 \rfloor$-th row from this ordering, yielding 40 held-out groups and 165 training groups.

The L3 to L5 training/holdout split contains 205 total examples: 65 Level 3, 79 Level 4, and 61 Level 5 problems. The balanced evaluation set is independent of the training split and is drawn from the MATH test set using 40 examples per level. The balanced evaluation set is used for the main difficulty-tier table.

D Prime Lab Reward-Hacking Experimental Configuration

Environment configuration. The sweep uses the `aggall` configuration with `inoculate=restrict`. In this setting, the visible task aggregates all visible constraints and the prompt includes an inoculation/restriction against metallic colors. We set `hidden_weight=0.0` in the environment so that the hosted environment's scalar reward is not directly optimized as a hidden-reward sum. Instead, the custom advantage modes expose the visible reward as the primary outcome signal and the hidden reward as the auxiliary signal for GDPO or SSAC. Thus hidden reward is both a monitored hacking metric and the auxiliary pressure in the custom advantage construction.

Field	Value
Provider	Prime Lab hosted training
Environment ID	`prime/backdoor-ifeval-all`
Environment family	Backdoor IFEval
Aggregation	`all`
Inoculation	`restrict`
Visible reward	Deterministic instruction-following score
Hidden reward	Trigger objective: `silver`
Environment hidden weight	0.0
Policy model	Llama-3.2-1B-Instruct

Table 8: Prime Lab adversarial reward-hacking environment configuration.

E AI Usage

Coding was done using Codex by OpenAI and manually verified. Writing was criticized using Claude and written and rewritten manually.