Islamic University Journal of Applied Sciences (IUJAS)

https://journals.iu.edu.sa/jesc

Volume VIII, Issue II, December 2026, Pages 346-366

Form over Substance: A Metric-Decomposition Diagnosis of Group Relative Policy Optimization for Arabic Legal Reasoning

Nawaf Al Enezi [1, *], Abdullah Alshanqiti [1], Ahmad B. Alkhodre [1]

[1] Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, KSA

451027326@stu.iu.edu.sa ; amma@iu.edu.sa ; aalkhodre@iu.edu.sa

*Corresponding author: Nawaf Al Enezi, *Email Address*: **451027326@stu.iu.edu.sa**

Abstract

Reinforcement learning with verifiable rewards (RLVR) drives much post-training of reasoning models, yet it is judged by a rising reward curve that cannot indicate which of several rule-based verifiers earned the gain. This study makes that observable during training through the weight-zero monitor: the substantive verifier is registered a second time at weight 0.0 and logged alongside the within-group standard deviation of each component, so that it cannot be optimized against. It was applied within an internally registered falsification program of Group Relative Policy Optimization (GRPO) on Qwen3-8B over Saudi commercial rulings. The registered attempt moved the policy without improving it: every greedy completion changed, 54% of held-out dispositions changed, macro-F1 fell by 0.067 (95% CI [−0.136, +0.011]), and repetition halved while the monitor remained flat. Acting on this reading, eight-sample majority voting, which trains nothing, achieved the only significant gain, +0.088 macro-F1 on a frozen 300-case instrument (95% CI [0.043, 0.140]). We derive when a gated composite reward rewards form rather than substance, and we recommend measuring the extractable headroom before allocating compute to policy-gradient training.

Keywords: Arabic legal natural language processing; reinforcement learning with verifiable rewards; group relative policy optimization; reward decomposition; legal judgment prediction; self-consistency.

https://doi.org/10.63070/jesc.2026.046

Received 20 August 2026; Revised 2 October 2026; Accepted 10 October 2026.Available online 16 October 2026.

Published by Islamic University of Madinah on behalf of *Islamic University Journal of Applied Sciences.*

This is an open access article published under the Creative Commons Attribution 4.0 International license (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/

1. Introduction

1.1 Problem statement

Group Relative Policy Optimization (GRPO) replaces the learned value function of proximal policy optimization with a group-relative baseline, and consequently it made reinforcement learning with verifiable rewards (RLVR), i.e., policy-gradient training against deterministic rule-based verifiers, affordable at post-training scale [1], [2]. The standard evidence that such a run has worked is a rising aggregate reward curve. When the reward is a composite (i.e., a weighted sum of several verifiers), this evidence is unsound, since the sum allows easily saturated components (e.g., a format check or a repetition check) to accumulate reward that the substantive component, namely the correct answer, never earns. Throughout this paper, we call form what a verifier can check without reading the case (the well-formedness of the output, its length, its repetition rate) and substance the operative disposition itself, i.e., the quantity the reward exists to improve. This is Goodhart's law [3] in an unusually clean form, as the proxy is an arithmetic sum of verifiers that are each correct on their own, and hence there is no learned judge whose errors could be audited. The parties affected are practitioners who post-train with composite verifiable rewards, most acutely in low-resource domains such as Arabic legal reasoning, where compute budgets are small and a misinterpreted run is costly. Such a practitioner needs a readout during the run, and none of the usual sources provides one: held-out evaluation is expensive and infrequent, the per-step reward is an aggregate and thus ambiguous, and the literature that motivates per-component logging [4] does not indicate whether a particular run is acquiring substance or form.

In this paper, we consider Arabic legal judgment prediction, i.e., predicting the operative disposition of a ruling from the case record, over Saudi first-instance commercial rulings. The published record on this task documents class imbalance and weak minority-class performance [5]–[7], reports that the customary automatic metrics are unreliable in this jurisdiction [8]–[11], and includes field-level critiques finding that much of the field does not measure what it claims to measure [12], [13]. Two properties of that record bear on our instrument. First, the disposition space is dominated by a single class, so a reward term that rewards surface form is easily satisfied while a class-balanced score such as macro-F1 remains unchanged. Second, the substantive endpoint (i.e., which disposition the court reached) already has an exact verifier, so reading it a second time costs nothing. To our knowledge, no existing instrument reports a property of one run at one step, namely which component is being learned now; this paper addresses that gap.

1.2 Contributions

The program reported here was designed to raise the held-out disposition accuracy of a distilled Qwen3-8B student with GRPO, beyond the level that supervised distillation had reached. It did not: the policy changed, but the disposition did not improve. This paper reports the instrument that made that failure observable while it was occurring, the diagnosis it supports, and the decision rule that recovered a gain without training. The originality and main contributions of this paper are outlined as follows:

- We introduce the weight-zero monitor, which duplicates the verifier of the substantive endpoint, computes it on every rollout, and registers it in the objective at weight 0.0, so that it

cannot be optimized against while measuring exactly the quantity the objective is meant to improve. It is logged alongside the within-group standard deviation of every reward component, at the cost of one deterministic function call per rollout.

- We report a diagnosis of the registered run, in which the objective was the disposition verifier itself with three monitors logged at weight zero. The logged aggregate did not rise, the substance signal stayed flat, and the held-out disposition score fell, while repetition halved and completions shortened. The movement, in other words, went into the quantities rewarded by the degenerate-group tie-break. That reading is consistent with the traces rather than established by an ablation.
- We propose a decision rule. Rather than iterating on the reward design, the program measured whether the capability being optimized for was already present in the sampling distribution (a quantity we call extractable headroom) and selected the next intervention accordingly; this selection produced the only significant improvement of the program, from an intervention that updates no parameters. The move has precedents [14]–[16], whose predictors are pre-hoc and aggregate; our monitor is in-training and per-component, and it answers a question they cannot, namely which component of a composite is being rewarded while the run is in progress.

1.3 Scope and paper organization

Regarding scope, our claims apply to the monitoring protocol and the decision rule, not to the accuracy levels observed (one court, one jurisdiction, one base model, and one reward family). The null reported here is therefore a null about this program rather than about verifiable-reward training in general.

2. Related Work

This section states, for each body of work our design rests on, what it settles and what it leaves to us: how composite rewards fail under group-relative optimization (Section 2.1), how such failures are detected while a run is in progress (Section 2.2), and what the Arabic and Saudi legal judgment prediction literature establishes about our testbed (Section 2.3). Our reading is that the first explains failures after the fact, the second is built for learned reward models rather than deterministic verifiers, and the third has met neither.

2.1 Composite rewards under group-relative optimization

GRPO samples a group of completions per prompt, scores each with the reward, and uses the within-group standardized reward as the advantage, dispensing with the learned value function of proximal policy optimization [1], [2]. Reward variance within a group is therefore a precondition for learning rather than a side effect: the expected policy gradient vanishes when the reward standard deviation is small even when the expected reward is far from optimal [17], and under binary rewards the fraction of degenerate groups exceeds the independent-Bernoulli prediction [18]. Entropy collapse and degenerate groups are documented at scale [19].

The line closest to our own concerns what happens when the reward is a sum of several verifiers. GDPO shows that normalizing after aggregation destroys component identity, collapsing distinct reward combinations into identical advantages, and that normalizing each component by its own

standard deviation restores each to exactly its weight [4]; MO-GRPO reaches the same repair for multi-objective settings [20]. The diagnosis has since been sharpened: MDP-GRPO names three separate pathologies of z-score normalization under discrete low-dispersion rewards, including zero-variance collapse [21], and a local covariance law states the condition under which an objective improves at all [22]. Against that, format and length signals alone have been shown to dominate early training and to account for most of the apparent gain [23], which is the specific risk our instrument is built to read. We do not read this literature as unanimous. Against the collapse narrative, standard-deviation normalization has been shown to implement an adaptive gradient with a provably improved convergence rate over unnormalized REINFORCE, governed by the average within-prompt reward standard deviation [24]: the mechanism that destroys resolution in a composite is what makes GRPO converge quickly when variance is healthy, which is why Section 5.3 treats the gate as a variance source rather than a defect.

Two further results bear on the outcome we report. First, the coupling between supervised fine-tuning and subsequent reinforcement learning: the two are proved not to decouple in either order, with the predicted degradation confirmed on a Qwen3 model [25]; checkpoints that have received excessive supervised fine-tuning lose the plasticity that policy-gradient training needs [26], while on-policy reinforcement learning prefers the Kullback–Leibler (KL)-minimal solution among those that solve the task [27]; and GRPO has been characterized as amplifying capabilities a model already has where supervised fine-tuning replaces them [28]. A distilled parent is the configuration these results warn about. Second, a deflationary reading that a reviewer may fairly raise against any GRPO-specific claim: under the standard structural assumptions, GRPO reduces to filtered iterative supervised fine-tuning, which matches it empirically across model families [29]. We do not adjudicate it here; it would not change our null, only its name.

Running against all of the above is a positive result in the setting that ought to have favored ours: deterministic rule-based verifiers applied to intermediate reasoning steps in a guideline-defined domain report up to 20% higher F1 than the prior state of the art and 6.5% over verifiable outcome rewards [30]; Section 5.4 takes that counter-case seriously.

2.2 Detecting reward hacking while a run is in progress

That a detection signal must be held outside the objective is established rather than assumed. The proxy score of a learned reward model keeps rising after the gold score it approximates has peaked [31]; folding a chain-of-thought monitor into the training reward yields obfuscated reward hacking, so a monitorability tax must be paid [32]. A recent survey organizes detection and mitigation by what they intervene on [33], and empirical work supplies invariance-based stress tests with early-warning signals [34].

What this literature does not supply, to our knowledge, is a per-component readout of a live run whose verifier is exact. Its instruments are built for learned reward models, where the monitor is itself a model. The nearest relative is a diagnostic rather than a monitor: the Advantage Collapse Rate quantifies the proportion of batches with ineffective gradients and predicts training stagnation [35]. We regard our contribution as extending that idea from the aggregate to the component, in the deterministic case where the endpoint already has an exact verifier and reading it again costs one function call per rollout.

We note one disagreement our result speaks to. One line argues that adding format or process components to the reward suppresses hacking, in medical question answering [36] and through process-outcome consistency filtering [37]; another finds that such components capture the gradient and bias the policy toward surface properties, with static-analysis rewards shortening completions without improving functional correctness [38]. When a format component is a safeguard and when it is the hack is, in our reading, unresolved; Section 5.3 offers a partial criterion.

2.3 Arabic and Saudi legal judgment prediction

Legal judgment prediction is surveyed in [5] and, for the legal domain more broadly, in [39]; LEXTREME and LexGLUE frame the task and its metrics [6], [7]. For Arabic the record is recent and concentrated. Ammar et al. benchmark fourteen model variants on 10,813 Saudi commercial cases and report that every automatic metric except human evaluation was inconsistent [8]; Kmainasi et al. build an Arabic dataset from the same court and show that low-rank adaptation (LoRA) fine-tuning of small models matches larger ones [9]; Almalki et al. reach 85.62% on a binary accept-or-reject formulation over 19,822 cases with sentence-level AraBERT and aggregation [10]; and ALARB adds 13K cases with court reasoning and cited clauses, demonstrating instruction tuning to near-frontier verdict prediction [11]. Alongside these sit an Arabic legal knowledge benchmark [40] and further Saudi commercial studies [41], [42].

Two findings from this literature shape our evaluation rather than merely preceding it. The first is critical: a substantial part of the field has been argued not to measure what it claims to measure [12], [13], and this has been quantified elsewhere in a low-resource language, where the model with the highest accuracy on an imbalanced legal task was a majority-class predictor scoring 23% macro-F1 while the genuinely best model scored 44% [43]. Our disposition space is dominated by one class, which is why macro-F1 and not accuracy is the criterion throughout this paper. The second concerns reproducibility: standardized evaluation of reasoning claims shows most reported reinforcement-learning gains to be far smaller than claimed, with seeds, decoding parameters, and prompt formatting accounting for much of the difference [44]. We report a null, so it does not threaten our result; it sets the standard our reporting tries to meet.

The literature therefore establishes that composite rewards fail through normalization and gate effects, that the repairs are per-component, that a detection signal must stay outside the objective, and that Arabic legal judgment prediction is dominated by a single class and poorly served by its customary metrics. What it does not provide is a per-step, per-component readout of a live run with an exact verifier, nor a registered program that adjudicates GRPO on this task with class-balanced scores and rules fixed in advance; those two gaps determine Sections 3.2 and 3.3.

3. Methods

3.1 Experimental program and evaluation instruments

The program was specified before execution as a set of single-variable interventions, or arms, branched from one parent checkpoint. Each arm varies exactly one factor, and each is scored on the primary evaluation set by the same rule-based disposition classifier that produced the reference labels. Nine paired evaluations across five intervention families were adjudicated: the registered final attempt, an

independently authored conservative recipe, six further optimizer families, and one arm that changed the learning rate alone; a first probe under corrected numerical settings preceded them as an instrument check and is reported in Section 4.3 but not counted. Every arm carried its decision rule in advance.

The design separates two questions that a single run confounds. The first is mechanical: did the optimizer update the model at all? The second is substantive: did the update change what the model decides on cases it has not seen? Table 2 reports both for every arm, and that separation is what makes a null readable. The student is Qwen3-8B [45], adapted with LoRA [46], in which small trainable matrices are added to the frozen base weights, trained with TRL's GRPO trainer (TRL 0.29.1, PEFT 0.19.1 (parameter-efficient fine-tuning), Transformers 5.12.1, PyTorch 2.11.0) and evaluated through vLLM 0.25.1 [47]; Table 3 lists the configuration. The adapter is a rank-64 LoRA on every linear projection, the condition under which the learning-rate analysis of Section 5.2 can rely on [48]. The corpus is split at the level of the case, so no ruling contributes text to more than one split, and the held-out split is never trained on and never used for checkpoint selection. Prompts longer than 3,072 tokens are dropped rather than truncated, so that no training prompt carries a partial case record; approximately 20,000 unique prompts remain for the optimization stage. A structural audit over the line found fully well-formed output, with mean coverage of the reasons and ruling sections of 0.966 and 0.946 against the source documents. Table 1 collects the data quantities at each stage.

Table 1. Data quantities at each stage. Splitting is at the level of the case, so no ruling contributes text to more than one split.

Stage	n	Role
Distillation traces	1,626	Verified chain-of-thought supervision for the parent checkpoint
Optimization prompts	≈20,000	Policy-gradient stage, after the 3,072-token length filter
Adjudication slice	200	Per-arm paired evaluation (training-lever family); drawn from the held-out split
Frozen instrument	300	Inference-composition family; held out from all training
External comparison set	200	Frontier contrasts; leakage-audited

3.2 The weight-zero monitor

A weight-zero monitor is an ordinary reward function registered with the trainer at weight 0.0: it is computed on every rollout and logged at every step, and it contributes nothing to the advantage, so no optimization pressure can corrupt it and it measures without bias the quantity the composite is intended to improve. The per-component within-group standard deviations sharpen the reading at the same cost: for a binary verifier with success probability p, the within-group standard deviation is $\sqrt{(p(1-p))}$, so a component that is being learned shows up as a collapse of its variance, and one that is being ignored as a flat band. Three properties matter: the monitor must be the endpoint's own verifier rather than a proxy; it must be computed on every rollout from the first step rather than at checkpoints; and it must be logged beside the per-component standard deviation, or a flat monitor is consistent with several mechanisms and identifies none. TRL already logs the mean and standard deviation of every registered reward function and accepts a weight of 0.0 [49]; the contribution is the prescription for using that telemetry. Throughout, we use exactly two terms for this instrument and its output: the weight-zero monitor is the registered verifier itself, and the substance signal is the reading it logs.

Before trusting the monitor, we characterized it off-line on stored rollouts (see Fig. 1). *Calibration.* With the fraction of correct completions in a group held at known values, the monitor's mean reproduces that fraction exactly; the within-group standard deviation is not monotone, peaking at 0.500

when half the group is correct and returning to zero at both extremes, so a reading of 0.433 is consistent with a success probability of 0.25 or 0.75, and direction is read from the mean. *Resolution.* At the 16 completions per update this program used, a single reading near a success probability of one half carries a standard error of 0.124; detecting a true shift of 0.10 requires averaging over roughly 25 steps, and a shift of 0.05 roughly 99 steps, so the 100-step arms can see a large movement in substance and not a small one, and the null of Section 4.2 is a null at that resolution. *Controls.* Permuting the reference labels 4,000 times drops the monitor from 0.450 to 0.192 (95% interval [0.157, 0.230]), so it reads the verifier rather than its own sampling variation; applied to two checkpoints whose capability differs by construction, the distilled parent and the same student after targeted verified injection, it moves in the correct direction, from 0.4500 to 0.5067, a paired difference of 0.0567 with an interval of [−0.0033, 0.1167]: it resolves checkpoints separated by about a tenth and not by a twentieth. Read on the training batch, the step-to-step series is over-dispersed relative to a constant-ability model (permutation p = 0.001) with no trend (p = 0.93): with two cases per update, the reading reflects which cases were drawn, so the monitor should be computed on a probe set held fixed across steps, as Section 5.5 states.

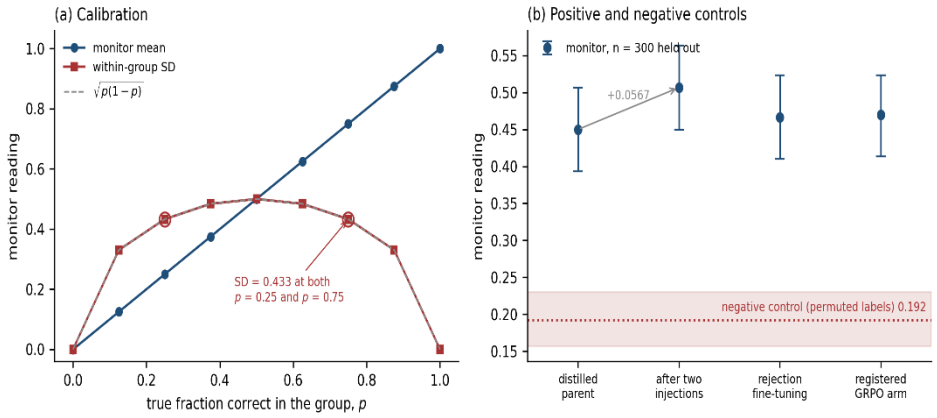


Fig. 1. Validating the weight-zero monitor off-line. (a) Calibration against a known fraction correct: the mean is exact; the within-group standard deviation is non-monotone. (b) Controls on the frozen 300-case instrument: permuting the reference labels collapses the reading to the shaded band, and the monitor separates the distilled parent from the injection-augmented student in the correct direction with an interval that touches zero.

3.3 Data, reward design, and falsification protocol

Our parent checkpoint is a Qwen3-8B student distilled on 1,626 verified chain-of-thought traces over published Saudi first-instance commercial rulings, in the reasoning-distillation line that transfers reasoning from a larger teacher [50]. Each trace was admitted only after clearing deterministic acceptance checks against the source document. Rewards are pure functions of the completion and of the gold verifiable fields: deterministic, GPU-free, and unit-tested. The objective actually optimized in the registered attempt is a single binary driver with three weight-zero monitors: $R = 1.0 \cdot D + 0.0 \cdot M_{\text{dec}} + 0.0 \cdot M_{\text{amt}} + 0.0 \cdot M_{\text{deg}}$. The driver D reads the disposition alone: a completion that fails the format gate (one reasoning block, the required section tags in order, a non-empty operative section) scores −0.0725; one that passes scores $(r - \rho_0)/(1 - \rho_0 - \rho_1)$, where $r \in \{0, 1\}$ records whether the extracted disposition matches the reference and $\rho_0 = 0.02$, $\rho_1 = 0.09$ are the audited reference-label noise rates, so a correct disposition scores 1.101, an incorrect one −0.022, and the gate score lies 0.05 below that of an incorrect answer. When all eight completions of a group score alike, a tie-break of at most 0.1 is

added, 0.06 for the fraction of reference amounts recovered and 0.04 for low repetition; the recorded degenerate-group rate was 0.19. The amount verifier enters the objective only through that tie-break, and the partial-credit driver analyzed in Section 5.3 (−0.5 behind a closed gate, $0.3 + 0.7 \cdot c$ behind an open one) is a design the program specified and no recorded arm ran; the structural partial-credit arm of Table 2 changed only the degenerate-group tie-break, not the driver. The monitors are the disposition match (M_{dec}), the fraction of reference amounts recovered (M_{amt}), and a binary degeneration check (M_{deg}: at least 20 tokens, no token above 30% of the text, Latin at most 5% of letters, CJK at most 0.5% of characters). Table 3 gives the trainer configuration.

Table 3. Configuration of the registered attempt, as recorded in its trainer arguments and adapter configuration.

Setting	Value
Optimizer steps	100
Learning rate	1×10^{-4} peak, linear decay to zero, no warm-up (logged 9.6×10^{-5} at step 5 to 1.0×10^{-6} at step 100)
Divergence penalty	none ($\beta = 0$)
Optimizer	AdamW, no weight decay, gradient clipping at 1.0
Rollouts	8 completions per prompt, 2 prompts per generation batch, 2 optimization passes per batch
Loss	DAPO (decoupled clip and dynamic sampling policy optimization) token-level; clipping 0.2 below, 0.4 above; group-mean advantage without standard-deviation scaling
Sampling	temperature 1.0, top-p 0.95, at most 2,048 tokens, truncated completions masked
Reward weights	[1.0, 0.0, 0.0, 0.0] over (D, M_{dec}, M_{amt}, M_{deg})
Adapter	LoRA, rank 64, $\alpha = 128$, no dropout, on q, k, v, o, gate, up, and down projections
Precision and seed	bfloat16; seed 42

Prompts were drawn from a success band of 0.25 to 0.75 under the parent policy, so that within-group reward variance would be non-zero by construction, which is the precondition for a non-vanishing policy gradient [17]. The run used a group size of 8, with two optimization iterations per generation batch, at sampling temperature 1.0, in bf16 precision, on a single A100 GPU. Its budget was fixed in advance at 100 steps: with two prompts per step and eight completions each, the registered attempt consumed 200 prompts and 1,600 rollouts, approximately 11% of the difficulty band (the trainer recorded 0.107 of an epoch at step 100) and about 1% of the length-filtered pool.

3.4 Intervention arms and external comparison

Five factors were varied one at a time: the numerical settings, the divergence anchor (the KL-penalty coefficient that ties the policy to the reference), the optimizer family, the learning rate, and the step count. Each arm branched from the same parent, was scored on the same evaluation instrument by the same frozen classifier and carried its adjudication rule before it ran. In addition to these training-lever arms, two further families change no parameters and are scored on the frozen 300-case instrument. The first quantifies extractable headroom by comparing eight-sample majority voting, in the self-consistency tradition [51], with greedy decoding, recording mean modal agreement alongside. The second composes voting with retrieval: three precedents per case retrieved by BM25, the lexical baseline of [52], over an index built from the training split only, in the retrieval-augmented tradition [53] and following the precedent-enhanced prompt template [54]; the same system is also decoded greedily so that the increment of the composition over voting alone can be tested. The delivered system is that composition, with no policy-gradient component, run in two independently seeded serving sessions so that sampled-serving variation could be reported as a range. The external comparison reads

it against Gemini 3.1 Pro (google/gemini-3.1-pro-preview) and GPT-5.5 (openai/gpt-5.5) on a 200-case set drawn at a fixed seed from a leakage-audited population disjoint from all training data, audited in the manner urged by [55], with identical prompts, candidate pools, and metrics at temperature 0 and pinned model identifiers. Because the frontier systems are not trained on our tag schema, a fallback that classifies the tail of the free text was added for the comparators and not for our own arms, an asymmetry that runs in the comparators' favor.

3.5 Statistical protocol

Three evaluation sets serve the paper, and each analysis names its set: the fixed 200-case held-out slice carries the nine paired arm evaluations of Section 4.1; the frozen 300-case instrument carries the parameter-free inference-composition results of Section 4.4 (majority voting, the precedent-augmented composition, and their increment); and a separate 200-case leakage-audited set carries the external comparison of Section 4.5.

All headline comparisons in this paper are paired. Metrics receive 95% intervals from 10,000 seeded bootstrap resamples [56]; McNemar's test [57] adjudicates paired per-case correctness with exact mid-p values; any interval containing zero is inconclusive whatever the point estimate. The Holm-Bonferroni correction [57] is applied within three declared contrast families and never across them: the training-lever family (the nine paired evaluations of Section 4.1 on the 200-case slice), the inference-composition family (voting against greedy decoding, voting with three precedents against greedy decoding, and the increment of the composition over voting, on the frozen 300-case instrument), and the external-comparison family (three pairwise system contrasts at n = 200). The resolving limit was stated before any result: at n = 300 and a discordance of 10 to 20%, the paired accuracy half-width is approximately ±0.036 to ±0.051. That proved too narrow for independently trained systems, where realized discordance was 37 to 38%, and on the 200-case external comparison the limit expands to approximately ±0.084. Decision rules were entered before the run each governed, in a dated change-control record kept with the project, following the practice described in [58]. Intervals are reported in one format throughout: the difference first, then the bracketed 95% interval, then the sample size and test, as in Δ macro-F1 = −0.067 (95% CI [−0.136, +0.011], n = 200, paired bootstrap); McNemar outcomes are reported as discordant counts with their mid-p.

4. Results

4.1 Overview of the falsification program

Table 2 shows, for every arm, the evidence that the weights moved, the share of held-out decisions that changed, and the adjudicated change in macro-F1, and Fig. 2 plots the four single-run arms against the parent checkpoint. As can be observed, no interval excludes zero.

Table 2. The falsification program: each arm, the evidence that its weights moved (ΔLoRA, the maximum absolute change in any adapter parameter), its behavioral change on held-out cases, and its adjudication; steps are completed/configured.

Arm	Steps	Evidence that weights moved	Held-out decisions changed	Δ macro-F1 (95% CI)	Adjudication
Corrected probe	85/100	max |ΔLoRA| 3.546×10^{-3}	0%	0.000 [0, 0]	Weights moved; greedy policy frozen

Arm	Steps	Evidence that weights moved	Held-out decisions changed	Δ macro-F1 (95% CI)	Adjudication
Registered final	100/100	max |ΔLoRA| 3.78×10^{-3}	54% of 200	−0.067 [−0.136, +0.011]	Movement without gain
Conservative standalone	100	max |ΔLoRA| 4.35×10^{-4}	Changed	+0.026 [−0.029, +0.081] on slice; frozen-instrument macro-F1 −0.015	Non-significant; frozen-instrument criterion fell
Six optimizer families	7 runs	Valid meters	Varied	No accepted gain in any	Locked null
Reduced learning rate 3×10^{-5}	100	|Δ| tracks the rate	88 of 200	−0.002 [−0.071, +0.081]	Damage removed; no gain beneath it

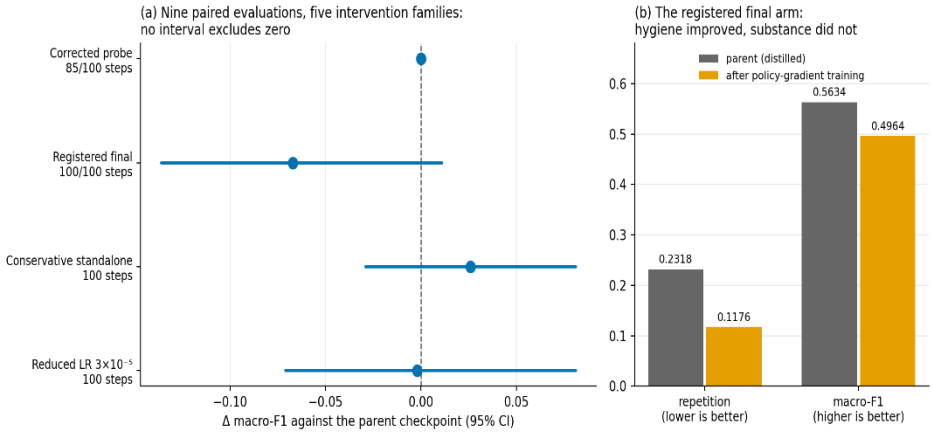


Fig. 2. The falsification program against the parent checkpoint. (a) Change in macro-F1 with 95% bootstrap intervals for the four single-run arms of Table 2, each scored on the fixed 200-case slice; no interval excludes zero. (b) The registered final arm: the repetition rate falls while macro-F1 also falls.

4.2 Registered attempt: policy change without improvement

The registered final attempt ran on rollouts certified clean by the instrument-repair campaign of Section 4.3 and completed its hundred steps at full update scale. Specifically, it moved: every one of the 200 greedy completions differed from the parent's, and 108 of them (54%) carried a different operative ruling. It did not improve: accuracy fell from 0.505 to 0.455 and macro-F1 from 0.5634 to 0.4964, a difference of −0.067 whose 95% bootstrap interval is [−0.136, +0.011] and spans zero; McNemar's test returned 36 against 26 discordant cases at mid-p = 0.21, and the gap was unchanged under alias-tolerant extraction (0.465 against 0.520). One thing improved, and it is form: the repetition rate halved, from 0.2318 to 0.1176, although repetition entered the objective only through the degenerate-group tie-break, which supplied the whole advantage in the 19% of groups whose dispositions scored alike; the substantive term, which the other groups optimized directly, deteriorated in the same run.

A second recipe, authored independently and branched from the same parent, moved the same slice the other way: accuracy rose by 0.035 and macro-F1 by 0.026, with an interval of [−0.029, +0.081] and McNemar p = 0.28; on the frozen 300-case instrument its raw accuracy rose while its macro-F1 fell by 0.015, the pattern expected when predictions shift toward the largest class. Six further optimizer families, run as seven single-variable arms (one family contributed two runs differing only in seed), produced no accepted gain; ranked by point estimate, the arms that concentrated or reshaped the gradient most aggressively sit at the worst end of the range, an ordering that is descriptive only and not evidence of a relationship, since every arm is single-seed and the spread across arms confounds

the intervention with training variance. The last arm changed one number, the learning rate, reduced to 3×10^{-5}, and the loss shrank with it to -0.002 with an interval spanning zero.

4.3 Alternative explanations excluded by the instrument

A null is admissible only under a design that excludes the alternatives, and here each exclusion is a measurement rather than an argument. *Defective machinery.* The earlier probe moved weights by 3.546×10^{-3} and changed 0% of decisions, the signature of a live optimizer behind a frozen generation path; it stopped at 85 of 100 steps because of an environment fault. Seven instrument defects were closed against recorded diagnostic output before the registered attempt ran, and one earlier verdict, which had read positive on a movement of +0.180 in the weight-zero monitor, was retracted when the corrected re-run reduced it to 0% of decisions changed and a macro-F1 difference of 0.000 [0, 0]. *Advantage collapse.* The zero-variance group fraction read 0.0 at all 20 logged readings, against a run-level collapse rate of 0.07, whereas the documented failure regime runs from 0.28 to 0.45. *Ineffective updates.* Full-precision snapshots put the maximum absolute adapter delta at 3.78×10^{-3}, and the cryptographic hash of the step-100 adapter was verified against the canonical adapter. *The weight-zero monitor.* The substance signal stayed flat in every window: 0.431 over the first half of the run and 0.332 over the second, 0.382 over all twenty readings, on the training-batch series that Section 3.2 characterizes and Fig. 4(b) draws; the driver itself, at 0.459 and 0.350 over the same halves, moved with it. A substance signal that does not move while the form terms improve is the form-over-substance signature stated as a measurement, available at every step for one deterministic function call per rollout. *Configuration.* Nine factors were varied across the arms (base model, difficulty band, reward gate, precision, sampling regime, monitor calibration, learning rate, divergence anchor, and step count), and the held-out measurement did not change sign in any consistent direction.

4.4 Extractable headroom and majority voting

Priced as described in Section 3.4, the sampling distribution already contained correct dispositions that greedy decoding was discarding (i.e., the extractable headroom was positive). Eight-sample majority voting, which updates no parameters, produced the program's only significant gain: on the frozen 300-case instrument, accuracy rose from 0.450 to 0.550 and macro-F1 by 0.088, with an uncorrected 95% paired-bootstrap interval of [0.043, 0.140]. Within the inference-composition family of three, the most stringent Holm step tests at $\alpha = 0.0167$; a percentile re-run of the paired bootstrap at that level (10,000 resamples, seed 3407) gives [0.032, 0.162], which excludes zero and reproduces the registered point estimate and lower bound exactly. Mean modal agreement was 0.719 to 0.730, so identical prompts yielded disagreement among samples in roughly 27 to 28 percent of cases. Composing voting with three retrieved precedents reached 0.5667 accuracy and 0.5558 macro-F1, a macro-F1 gain of 0.1728 over the same system decoded greedily (0.3830); self-consistency was unchanged at 0.717 to 0.734, so the retrieval channel added information rather than stability. Across two independently seeded serving sessions the delivered system reaches 0.600 accuracy and a macro-F1 of 0.563 to 0.573, the range recording sampled-serving variation. Extractable headroom was negative before targeted verified injection, positive after the first injection, and saturated within session noise after the second.

4.5 External comparison

Under the protocol of Section 3.4, the composed system reached 0.630 accuracy and 0.6389 macro-F1, against 0.660 accuracy and 0.5712 macro-F1 for Gemini 3.1 Pro and 0.640 accuracy and 0.5664 macro-F1 for GPT-5.5. No pairwise difference in accuracy was rejectable at this sample size after Holm correction. We state this as far as the test licenses it and no further: non-rejection at a resolving limit of approximately ±0.084 is not measured equivalence, and we do not upgrade it to parity anywhere. The macro-F1 differences, which favor the composed system, were not tested: the declared external family of Section 3.5 holds the three per-case accuracy contrasts, and no post-hoc macro-F1 contrast was constructed. After the free-text fallback described in Section 3.4, the residual unscoreable rates were 0.5% for the composed system against 8.0% and 1.0% for the comparators; this is reported as an operational contrast rather than as a quality claim.

4.6 Training traces of the registered attempt

Fig. 4 presents the traces of the registered attempt, drawn from the trainer log as recorded. The gradient norm had a median of 0.485 and rose from 0.28 to 1.00 as the learning rate decayed; no divergence term was logged, because none was configured. The logged aggregate, the unweighted sum of the four functions, moved from 2.49 to a maximum of 2.84 and ended at 1.96, and the driver's batch mean drifted from 0.59 to 0.33. The degeneration term read 1.0 with zero spread throughout, the within-batch standard deviations of the driver and of the substance signal stayed between 0.22 and 0.57, and the mean completion length fell from 982 to 841 tokens. Under the constants of Section 3.3 the gate could not have carried the advantage in this attempt; where the disposition term had no variance, the tie-break did.

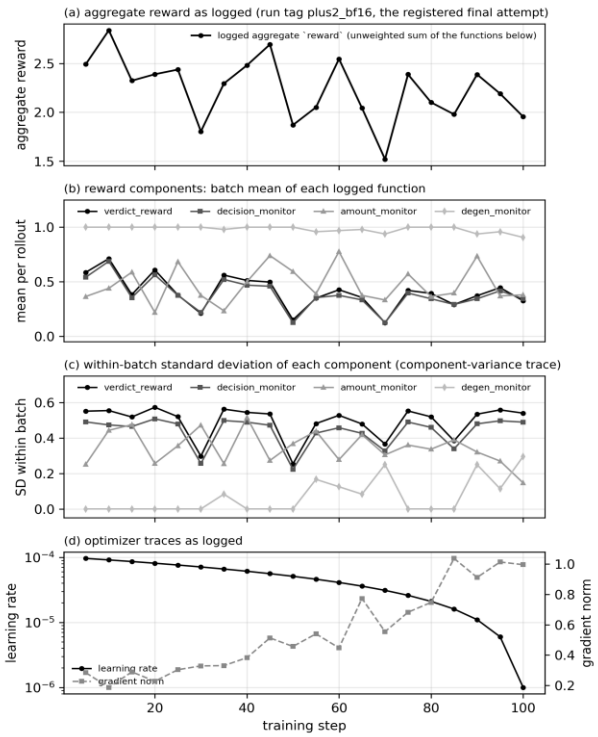


Fig. 4. Training traces of the registered final GRPO attempt on the Qwen3-8B student (100 steps, logged every five steps), drawn from the trainer log as recorded. (a) The aggregate reward as logged, the unweighted sum of the four reward functions. (b) Batch means of the four logged reward functions of Section 3.3: the degeneration term is saturated at 1.0 from the first reading, and the verdict reward and the substance signal fluctuate without trend. (c) Within-batch standard deviation of each function, the component-variance trace of Section 3.2. (d) Learning rate (left, logarithmic) and gradient norm (right).

5. Discussion

5.1 Principal findings and candidate causes

This paper reports three findings. The registered policy-gradient arm explored, and the objective could not rank that exploration into a gain on the substantive endpoint: the policy changed most of its held-out decisions and improved only on form, a repetition rate that halved and completions that shortened, the quantities rewarded by the degenerate-group tie-break. This is the form-over-substance pattern; an aggregate reward curve, had it risen, would have reported it as progress. The weight-zero monitor and its companion traces excluded defective machinery, advantage collapse, ineffective updates, and configuration as explanations while the run was in progress. And quantifying the sampling distribution instead of redesigning the reward showed that correct dispositions were already being sampled and discarded, so that majority voting, which trains nothing, produced the only significant gain, and composing it with retrieved precedents added information rather than stability, consistent with the documented sensitivity of generation to retrieved noise [59].

Several candidate causes are jointly consistent with the traces; the program was built to test two of them (Sections 5.2 and 5.3) and quantified a third, the absence of extractable capability, in advance. A fourth entered the literature after the program closed and has the standing of a proof: supervised fine-tuning and reinforcement learning do not decouple in either order, and reinforcement learning raises the supervised loss, with the predicted degradation confirmed on a Qwen3 model [25]. Our arms are exactly that configuration, and our held-out macro-F1 falls; three cautions attach, since the confirmation is at 0.6B parameters while this program runs at 8B, the theory bounds a direction rather than a magnitude, and the quantity it bounds is the supervised loss on the distillation traces, which this program did not measure. A fifth is related and equally untested here: a checkpoint that has received excessive supervised fine-tuning can lose the plasticity that policy-gradient training needs [26], and on-policy reinforcement learning prefers the KL-minimal solution among those that solve the task [27], which is consistent with a run that moved little from a rigid reference.

5.2 Effect of update scale and the divergence anchor

Two of the program's arms sit at opposite ends of the update-scale axis, and the contrast separates two nulls that a single run would confound. The conservative standalone recipe of Table 2 was configured by its independent author at a peak learning rate of 1×10^{-5} decayed linearly over its hundred steps and with a divergence penalty to the reference. Its log records a divergence with a median of 0.010, a gradient norm with a median of 0.25 that never trended, and an adapter whose largest entry changed by 4.4×10^{-4}, an order of magnitude less than the registered attempt's. The empirical analysis we rely on reports that the optimal learning rate for low-rank adaptation is about ten times the full-fine-tuning optimum, approximately invariant to the adapter rank [48]. It gives no absolute optimum for a configuration like ours, so the implied optimum of approximately 5×10^{-5} is our own inference from it; the conservative recipe's peak is five times lower than that value. Its anchor, moreover, ties the policy to a supervised checkpoint whose disposition accuracy is the 0.46 that the program never moved beyond: anchoring to a reference that lacks the target capability turns the regularizer into a constraint that holds the policy near the failure. The registered attempt is the other case: it ran at a peak of 1×10^{-4}, roughly twice the inferred low-rank optimum, with no divergence penalty, and it moved, the adapter's

largest entry changing by 3.78×10^{-3}, every greedy completion changing, and 54% of held-out dispositions flipping, while macro-F1 fell. Update scale therefore does not explain the registered null; it explains the conservative one. The arm that reduced the registered rate to 3×10^{-5} removed the loss without producing a gain beneath it, which identifies the mechanism of that loss as optimizer drift proportional to update magnitude rather than learning with the wrong sign; an independent controlled null on a supervised web-agent checkpoint reports the same gating by learning rate [16]. Two results bound what a policy-gradient arm can add once the margin is thin: reinforcing only correct responses improves pass@1 while degrading performance at higher k through lost diversity [60], and policy-gradient training with verifiable rewards has been characterized as reweighting existing solution traces rather than discovering new ones [61].

5.3 Advantage capture in gated composite rewards

Group-relative optimization standardizes each completion's summed reward within its group. Writing the standardized advantage as $A = (r - \bar{r})/\sigma_r$, where $r = \Sigma_c\, w_c r_c$ is the weighted sum of the components, each component's share of the direction of A scales with $w_c\sigma_c/\sigma_r$ when the components vary independently, so when one component's weighted standard deviation dominates σ_r the signal points almost entirely along it. This is the mechanism that GDPO [4] and MO-GRPO [20] repair by per-component normalization, that the covariance law of [22] states as the condition for an objective to improve, and that saturation-aware reweighting [62] addresses for components already solved. In the registered attempt the gate penalty was set at 0.05 (Section 3.3) and could not capture the advantage; form entered through the degenerate-group tie-break instead. The closed form that follows concerns the gated partial-credit driver, -0.5 behind a closed gate and $0.3 + 0.7q$ behind an open one, the common shape of composite verifiable rewards, and the one this program specified but never ran. It is a design result, and Fig. 3 reproduces it without any of our stack.

Write p for the probability that a completion is well formed and q for the probability that its disposition is correct. The law of total variance splits the driver's within-group variance into a gate term and a substance term, $\mathrm{Var}(D) = p(1-p)(0.8 + 0.7q)^2 + 0.49\, p\, q(1-q)$, and the two are equal only at $p^*(q) = 1 - 0.49\, q(1-q)/(0.8 + 0.7q)^2$, which is 0.903 at $q = 0.46$ (the disposition accuracy of the supervised reference, an illustrative value) and never falls below 0.898 for any q (Fig. 3b, minimum at $q \approx 0.35$). Below that threshold the gate carries more than half of the driver's variance and the standardized advantage is directed at format: the gap between well-formed and malformed completions is large, between 0.8 and 1.5 points, while substance, reachable only through an open gate, is worth at most 0.7. Fig. 3 reproduces the argument without project data: 400 sampled groups of sixteen Bernoulli draws match the closed form to within sampling error (maximum deviation 0.016 across the sweep), and nothing in the derivation refers to law, to Arabic, or to our corpus.

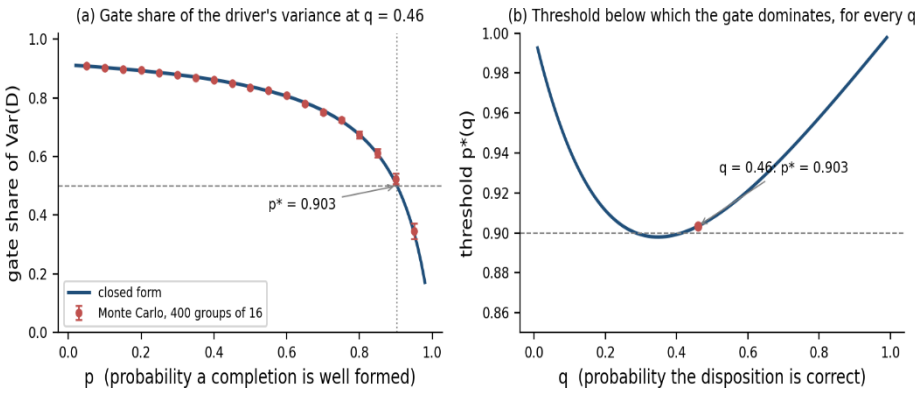


Fig. 3. The gate-variance argument reproduced without project data, using only the constants of the gated partial-credit driver stated in Section 5.3. (a) The gate's share of the driver's within-group variance as the well-formedness probability p is swept at $q = 0.46$, the disposition accuracy of the supervised reference: closed form (line) and Monte Carlo estimate (points, 95% bootstrap intervals) agree, and the share falls below one half only at $p^* = 0.903$, the equal-variance point. (b) The threshold $p^*(q)$: it lies between 0.898 and 1.0 for every q.

The gated design was our own engineering, built to prevent the pathology it invites, and it needs to be accounted for. Multiplying substance by a structure gate denies format reward without substance exposure, and under group normalization the gate then becomes the dominant variance source until nine rollouts in ten are well formed; the registered attempt avoided this by setting its gate penalty at 0.05 and still did not gain. One caveat qualifies the derivation: it treats components as separable, whereas in this design the gated driver is multiplicatively coupled to the structure gate by construction, so the covariance term is not zero and the decomposition is an approximation rather than an identity. The direction of the conclusion is unaffected, because coupling to the dominant-variance component strengthens rather than weakens its capture of the advantage.

5.4 Limitations and threats to validity

Causes that are jointly consistent with the traces constitute a consistency argument, not an ablation. One published result runs directly against the reading this section defends: deterministic rule-based verifiers in a guideline-defined domain are reported to outperform both the prior state of the art and verifiable outcome rewards [30], the favorable case our stack was supposed to be. We take the difference to be structural: that work verifies the process, scoring each intermediate step against the guideline that licenses it, whereas this stack verifies the outcome and gates it behind format, and Section 5.3 derives what gating costs. On that reading the counter-case supports the closed-form argument rather than contradicting the null; we have not run a process-verified arm on this corpus.

Training budget. Each arm saw 200 prompts and 1,600 rollouts, and the objection that the null is an artifact of too little data deserves a direct answer. First, the budget was sufficient to move the policy: every greedy completion changed, 54% of held-out dispositions changed, and the adapter's largest entry changed by 3.78×10^{-3} at a learning rate twice the inferred low-rank optimum, so the null is movement without gain rather than the absence of movement that under-training predicts. Second, Section 3.2 states the resolution at which the null is read: 100 steps resolve a shift of about 0.10 in the substance rate and not a smaller one, and the claim is bounded accordingly. Third, the volume of data is not the binding constraint when headroom exists; policy-gradient training with verifiable rewards has been shown to yield large gains from a single training example when the base policy already samples the correct answer [63], which is exactly the condition that Section 4.4 measured and found

exhausted for the registered parent. Whether a longer run on the full pool would behave differently is a question this program did not test, and Section 6 lists it. *Design and replication.* Every arm is single-seed, so breadth across nine paired evaluations substitutes for repetition within any one of them; small rollout groups leave exploration breadth open as an alternative account; and a full-fine-tuning control would test a different update-capacity hypothesis. A set of intervals that all contain zero remains a null however the variance is apportioned, whereas any ordering of the arms is not.

Model-family specificity. All arms use one base model, from a family on which gains from format-only or random rewards are documented and known not to transfer [64]. Part of that documented family effect has since been attributed to benchmark contamination: leakage-free generators show that only accurate reward signals produce steady gains [65]. A family-specific mechanism therefore cannot be excluded until the arms are repeated on a second base model.

Registration. Decision rules were not registered with a public service; they were entered, before the run each governed, in a dated change-control record in which every intervention carries a numbered change request, the single lever it varies, and the rule that adjudicates it.

Instrument. One rule-based classifier produced the reference labels and scores every arm, so any error in it moves both sides of every comparison. The version used was frozen before the registered attempt ran, and one documented bias, a character-ratio rule in the strict extractor that penalizes shorter outputs, was left in place rather than repaired after the fact.

Scope and reproducibility. Claims of generality apply to the monitoring protocol and the decision rule, not to the accuracy levels observed: one court, one jurisdiction, one base model, one reward family. The corpus cannot presently be redistributed pending a rights and anonymization review, which materially limits independent replication.

5.5 Recommended monitoring protocol and decision rule

Setting the specifics of our run aside, one instrument remains, and we recommend it to researchers training with verifiable rewards in general: weight-zero monitor rewards with per-component group-variance traces, logged from step zero. In our run, three such monitors cost no measurable compute and were the only instruments that distinguished a moving policy from a substance signal that barely moved. Before allocating compute to policy-gradient optimization, quantify the extractable headroom: measure whether the distribution the policy already samples contains correct answers that greedy decoding discards, a quantity that pass@k reads as a diagnostic [66]. Majority voting is itself a statistical certificate of the model's mode, and label-free post-training sharpens the distribution toward that same mode [67]; selection and training therefore compete for one margin rather than two. Selection rules admit sharper forms than plain voting: restricting candidates to high-frequency responses before ranking is minimax-optimal for pass@k inference scaling [68]. Register the verifiers that carry the objective a second time at weight zero, compute them on a probe set held fixed across steps, state the resolution at which their null is a null, and promote checkpoints only on held-out substance metrics. Where that headroom is exhausted, the effective intervention introduces new behavior into the sampling distribution, here targeted verified injection, in the manner of preparing a model for reinforcement learning by injecting behaviors before it [69].

6. Conclusion

This paper addressed the problem of reading a composite verifiable reward while it is being optimized. We have shown that such a reward can move a policy for a hundred steps toward shorter and less repetitive completions without affecting the quantity it was designed to improve, and that no aggregate readout reports this. Towards detecting it during the run, we have presented the least expensive instrument that does so, i.e., the endpoint's own verifier registered a second time at weight zero and logged with the within-group variance of each component. On an internally registered program of nine paired evaluations, this instrument produced a readable null (a registered arm that changed 54% of held-out dispositions while macro-F1 fell by 0.067 with an interval spanning zero) and directed the program to the one intervention that produced a gain, namely eight-sample majority voting, which trains nothing and raised macro-F1 by 0.088 (95% CI [0.032, 0.162] after correction). The claim we defend is narrow: where a substantive endpoint has an exact verifier, we recommend registering it twice, once inside the objective and once outside it at weight zero, so that the run can be interpreted; and the present results suggest quantifying the extractable headroom before policy-gradient optimization is undertaken. The findings are bounded by the limitations of Section 5.4: single-seed arms, one base model and reward family, 200 prompts per arm, and a corpus that cannot yet be redistributed. Given these limitations, future work will proceed in the following directions:

1. Multi-seed replication of the registered and extreme arms, besides larger rollout groups and longer runs over the full prompt pool.
2. Per-component normalization before the weighted sum [4], as a test of the mechanism derived in Section 5.3.
3. A plasticity diagnostic of the distilled parent checkpoint.
4. Repeating the arms on a second base model where the extractable headroom is positive rather than exhausted.

In conclusion, a rising reward curve is an indication rather than a measurement, and on the present results, registering the endpoint's own verifier at weight zero is the minimum instrumentation we would ask a composite-reward run to report.

Ethics Statement

This study involved no human subjects, human data, or personal data; the corpus derives from judicial rulings published as public records. The systems described are research artifacts and are not fit for autonomous adjudication.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability Statement

Three availability categories apply. (i) Not releasable: the implementation source code, whose export the authors' institution prohibits; as compensating disclosure, Section 3.3 states every reward function

and constant, and Section 5.3 derives the central mechanism in closed form so that it can be checked with algebra. (ii) Temporarily withheld pending review: the corpus, an independent collection assembled from rulings published as public records on the Saudi Ministry of Justice portal, and the trained adapters, both held until a rights and anonymization review concludes. (iii) Released with the article as supplementary material: the script that produces Fig. 3, which uses no project data; the script that produces Fig. 4; and the registered attempt's trainer log as recorded, a 39 kB text file that contains no case text.

Acknowledgement of AI Use

The authors used Claude (Anthropic) as an assistive tool for the following limited purposes: language refinement, structuring of certain sections, and preliminary organization of literature. All intellectual contributions, data interpretation, critical analysis, and final scientific judgments remain solely those of the authors. The AI was not used to generate original research ideas, fabricate data, or replace the authors' scholarly reasoning. The final manuscript was thoroughly reviewed and approved by the authors.

References

[1] Z. Shao et al., "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models," arXiv:2402.03300, 2024.
[2] D. Guo et al., "DeepSeek-R1 Incentivizes Reasoning in LLMs Through Reinforcement Learning," Nature, vol. 645, pp. 633–638, 2025, doi: 10.1038/s41586-025-09422-z.
[3] J. Karwowski, O. Hayman, X. Bai, K. Kiendlhofer, C. Griffin, and J. Skalse, "Goodhart's Law in Reinforcement Learning," in Proc. Int. Conf. Learn. Represent. (ICLR), 2024. arXiv:2310.09144.
[4] S.-Y. Liu et al., "GDPO: Group Reward-Decoupled Normalization Policy Optimization for Multi-Reward RL Optimization," arXiv:2601.05242, 2026.
[5] J. Cui, X. Shen, F. Nie, Z. Wang, J. Wang, and Y. Chen, "A Survey on Legal Judgment Prediction: Datasets, Metrics, Models and Challenges," arXiv:2204.04859, 2022.
[6] J. Niklaus, V. Matoshi, P. Rani, A. Galassi, M. Stürmer, and I. Chalkidis, "LEXTREME: A Multi-Lingual and Multi-Task Benchmark for the Legal Domain," in Findings Assoc. Comput. Linguistics: EMNLP, 2023. arXiv:2301.13126.
[7] I. Chalkidis et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," in Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL), Vol. 1: Long Papers, Dublin, Ireland, May 2022, pp. 4310–4330, doi: 10.18653/v1/2022.acl-long.297.
[8] A. Ammar, A. Koubaa, B. Benjdira, O. Nacar, and S. Sibaee, "Prediction of Arabic Legal Rulings Using Large Language Models," Electronics, vol. 13, no. 4, art. 764, 2024, doi: 10.3390/electronics13040764.
[9] M. B. Kmainasi, A. E. Shahroor, and A. Al-Ghraibah, "Can Large Language Models Predict the Outcome of Judicial Decisions?" arXiv:2501.09768, 2025.
[10] A. Almalki, S. Alsafari, and N. M. Alotaibi, "Legal Judgment Prediction in the Saudi Arabian Commercial Court," Future Internet, vol. 17, no. 10, art. 439, 2025, doi: 10.3390/fi17100439.
[11] H. Abu Shairah et al., "ALARB: An Arabic Legal Argument Reasoning Benchmark," in Proc. 3rd Arabic Natural Language Processing Conf. (ArabicNLP), Suzhou, China, Nov. 2025, pp. 389–406, doi: 10.18653/v1/2025.arabicnlp-main.32.
[12] M. Medvedeva, M. Wieling, and M. Vols, "Rethinking the field of automatic prediction of court decisions," Artif. Intell. Law, vol. 31, no. 1, pp. 195–212, 2023, doi: 10.1007/s10506-021-09306-3.

[13] M. Medvedeva and P. McBride, "Legal Judgment Prediction: If You Are Going to Do It, Do It Right," in Proc. Natural Legal Language Processing Workshop 2023, Singapore, Dec. 2023, pp. 73–84, doi: 10.18653/v1/2023.nllp-1.9.
[14] F. Kang, M. Kuchnik, K. Padthe, M. Vlastelica, R. Jia, C.-J. Wu, and N. Ardalani, "Quagmires in SFT-RL Post-Training: When High SFT Scores Mislead and What to Use Instead," arXiv:2510.01624, 2025.
[15] L. Zhang and J. Li, "Predicting Inference-Time Scaling Gains from Labeled Validation-Set Output Statistics," arXiv:2606.02981, 2026.
[16] C. Gan, Z. Cai, Y. Liang, H. Wei, S. Ni, and Q. Zhang, "A Learning-Rate-Gated Failure of GRPO in a Small Language and Vision-Language Model Web Agent: A Controlled Null and Its Mechanism," arXiv:2607.12640, 2026.
[17] N. Razin et al., "Vanishing Gradients in Reinforcement Finetuning of Language Models," in Proc. Int. Conf. Learn. Representations (ICLR), 2024. arXiv:2310.20703.
[18] W. Nie et al., "Gradient Starvation in Binary-Reward GRPO: Why Group-Mean Centering Fails and Why the Simplest Fix Works," arXiv:2605.07689, 2026.
[19] Q. Yu et al., "DAPO: An Open-Source LLM Reinforcement Learning System at Scale," in Adv. Neural Inf. Process. Syst. (NeurIPS), 2025. arXiv:2503.14476.
[20] Y. Ichihara et al., "MO-GRPO: Mitigating Reward Hacking of Group Relative Policy Optimization on Multi-Objective Problems," arXiv:2509.22047, 2025.
[21] M. M. Salmani-Zarchi, Z. Rahimi, H. Faili, and M. J. Dousti, "MDP-GRPO: Stabilized Group Relative Policy Optimization for Multi-Constraint Instruction Following," in Proc. 64th Annu. Meeting Assoc. Comput. Linguistics (ACL), Vol. 1: Long Papers, San Diego, CA, USA, 2026, pp. 42784–42797, doi: 10.18653/v1/2026.acl-long.1982. arXiv:2606.06058.
[22] Y. Lu and M. Jiang, "Uncovering Cross-Objective Interference in Multi-Objective Alignment," arXiv:2602.06869, 2026.
[23] R. Xin et al., "Surrogate Signals from Format and Length: Reinforcement Learning for Solving Mathematical Problems without Ground Truth Answers," arXiv:2505.19439, 2025.
[24] C. Ge, C. H. Yin, H. Liang, and J. Zhang, "Why GRPO Needs Normalization: A Local-Curvature Perspective on Adaptive Gradients," in 5th Workshop on Mathematical Reasoning and AI (MATH-AI), Neural Inf. Process. Syst. (NeurIPS), Dec. 2025. arXiv:2601.23135.
[25] X. Niu, B. Bai, W. Han, and W. Zhang, "On the Non-decoupling of Supervised Fine-tuning and Reinforcement Learning in Post-training," arXiv:2601.07389, 2026.
[26] R. Liu, J. Liu, X. Wan, Y. Fu, and L. Pan, "When RL Fails after SFT: Rejuvenating Model Plasticity for Robust SFT-to-RL Handoff," arXiv:2606.09932, 2026.
[27] I. Shenfeld, J. Pari, and P. Agrawal, "RL's Razor: Why Online Reinforcement Learning Forgets Less," arXiv:2509.04259, 2025.
[28] N. Rajani, A. P. Gema, S. Goldfarb-Tarrant, and I. Titov, "Scalpel vs. Hammer: GRPO Amplifies Existing Capabilities, SFT Replaces Them," in Actionable Interpretability Workshop, Int. Conf. Mach. Learn. (ICML), 2025. arXiv:2507.10616.
[29] S. R. Samineni, D. Kalwar, K. Valmeekam, K. Stechly, and S. Kambhampati, "RL in Name Only? Analyzing the Structural Assumptions in RL Post-Training for LLMs," arXiv:2505.13697, 2025.
[30] M. Pronesti, A. Belz, and Y. Hou, "Beyond Outcome Verification: Verifiable Process Reward Models for Structured Reasoning," in Findings of the Association for Computational Linguistics: ACL 2026, 2026. arXiv:2601.17223.
[31] L. Gao, J. Schulman, and J. Hilton, "Scaling Laws for Reward Model Overoptimization," in Proc. Int. Conf. Mach. Learn. (ICML), 2023.
[32] B. Baker et al., "Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation," arXiv:2503.11926, 2025.
[33] X. Wang et al., "Reward Hacking in the Era of Large Models: Mechanisms, Emergent Misalignment, Challenges," arXiv:2604.13602, 2026.
[34] I. F. Shihab, S. Akter, and A. Sharma, "Detecting Proxy Gaming in RL and LLM Alignment via Evaluator Stress Tests," arXiv:2507.05619, 2025.

[35] X. He et al., "Advantage Collapse in Group Relative Policy Optimization: Diagnosis and Mitigation," arXiv:2605.21125, 2026.
[36] M. F. Tarek et al., "Reward Hacking Mitigation using Verifiable Composite Rewards," in Proc. 16th ACM Int. Conf. Bioinformatics, Computational Biology, and Health Informatics (BCB), 2025.
[37] C. Ye et al., "Beyond Correctness: Harmonizing Process and Outcome Rewards through RL Training," in MATH-AI Workshop, Neural Inf. Process. Syst. (NeurIPS), 2025. arXiv:2509.03403.
[38] E. Skopin and E. Kotelnikov, "Improving Small Language Models for Code Generation with Reinforcement Learning from Verification Feedback," in Proc. Artificial Intelligence and Natural Language Conf. (AINL), 2026. arXiv:2605.30478.
[39] F. Ariai and G. Demartini, "Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges," ACM Comput. Surv., 2024.
[40] F. Hijazi et al., "ArabLegalEval: A Multitask Benchmark for Assessing Arabic Legal Knowledge in Large Language Models," in Proc. 2nd Arabic Natural Language Processing Conf. (ArabicNLP), Bangkok, Thailand, Aug. 2024, doi: 10.18653/v1/2024.arabicnlp-1.20.
[41] M. Altherwi, "The Efficacy of Large Language Models (LLM) in Predicting the Outcomes of Commercial Legal Cases based on The Saudi Law: A Preliminary Study," in Proc. 6th Int. Conf. Computing, Networks and Internet of Things (CNIOT), 2025, pp. 1–5, doi: 10.1109/CNIOT65435.2025.11070426.
[42] N. Alangari, N. Alshenaifi, and H. Almuzaini, "Hierarchical Classification of Arabic Legal Cases Using Transformer Architectures and Large Language Models," Electronics, vol. 15, no. 14, art. 2988, 2026, doi: 10.3390/electronics15142988.
[43] V. Ovcharov, "UA-Legal-Bench: A Benchmark for Evaluating Large Language Models on Ukrainian Legal Reasoning," arXiv:2605.29170, 2026.
[44] A. Hochlehnert et al., "A Sober Look at Progress in Language Model Reasoning: Pitfalls and Paths to Reproducibility," arXiv:2504.07086, 2025.
[45] A. Yang et al., "Qwen3 Technical Report," arXiv:2505.09388, 2025.
[46] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in Proc. Int. Conf. Learn. Representations (ICLR), 2022. arXiv:2106.09685.
[47] W. Kwon et al., "Efficient Memory Management for Large Language Model Serving with PagedAttention," in Proc. 29th ACM Symp. Operating Systems Principles (SOSP), 2023, pp. 611–626, doi: 10.1145/3600006.3613165.
[48] J. Schulman and Thinking Machines Lab, "LoRA Without Regret," Thinking Machines Lab: Connectionism, Sep. 2025, doi: 10.64434/tml.20250929. [Online]. Available: https://doi.org/10.64434/tml.20250929. Accessed: Aug. 4, 2026.
[49] Hugging Face, "TRL Documentation: Logging (GRPOTrainer logged metrics)," 2026. [Online]. Available: https://huggingface.co/docs/trl/main/logging. Accessed: Sep. 6, 2026.
[50] N. Ho, L. Schmid, and S.-Y. Yun, "Large Language Models Are Reasoning Teachers," in Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL), Vol. 1: Long Papers, Toronto, Canada, 2023, pp. 14852–14882.
[51] X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in Proc. 11th Int. Conf. Learn. Representations (ICLR), 2023.
[52] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in Proc. Neural Inf. Process. Syst. Datasets and Benchmarks Track, 2021. arXiv:2104.08663.
[53] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023.
[54] Y. Wu et al., "Precedent-Enhanced Legal Judgment Prediction with LLM and Domain-Model Collaboration," in Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP), Singapore, 2023, pp. 12060–12075.

[55] S. Balloccu, P. Schmidtova, M. Lango, and O. Dusek, "Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs," in Proc. 18th Conf. European Chapter of the Association for Computational Linguistics (EACL), Vol. 1: Long Papers, St. Julian's, Malta, Mar. 2024, pp. 67–93, doi: 10.18653/v1/2024.eacl-long.5.
[56] D. Ulmer, C. Hardmeier, and J. Frellsen, "deep-significance: Easy and Meaningful Statistical Significance Testing in the Age of Neural Networks," in Proc. ML Evaluation Standards Workshop, Int. Conf. Learn. Represent. (ICLR), 2022. arXiv:2204.06815.
[57] S. Riezler and M. Hagmann, Validity, Reliability, and Significance: Empirical Methods for NLP and Data Science, 2nd ed. Cham, Switzerland: Springer, 2024.
[58] E. van Miltenburg, C. van der Lee, and E. Krahmer, "Preregistering NLP Research," in Proc. 2021 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT), 2021, pp. 613–623.
[59] F. Cuconasu et al., "The Power of Noise: Redefining Retrieval for RAG Systems," in Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, Washington, DC, USA, 2024, pp. 719–729.
[60] X. Zhu et al., "The Surprising Effectiveness of Negative Reinforcement in LLM Reasoning," arXiv:2506.01347, 2025.
[61] B. Bai, X. Wang, P. Ye, and T. Chen, "Learning to Explore with Parameter-Space Noise: A Deep Dive into Parameter-Space Noise for Reinforcement Learning with Verifiable Rewards," in ICML Workshop on AI for Math, Jul. 2026. arXiv:2602.02555.
[62] Y. Wang et al., "Learn What's Left, Not What's Mastered: Saturation Aware Advantage Reweighting for Multi-Reward Policy Optimization," arXiv:2608.16072, 2026.
[63] Y. Wang et al., "Reinforcement Learning for Reasoning in Large Language Models with One Training Example," arXiv:2504.20571, 2025.
[64] R. Shao et al., "Spurious Rewards: Rethinking Training Signals in RLVR," arXiv:2506.10947, 2025.
[65] M. Wu et al., "Reasoning or Memorization? Unreliable Results of Reinforcement Learning Due to Data Contamination," in Proc. AAAI Conf. Artificial Intelligence, 2026. arXiv:2507.10532.
[66] Y. Yu, "Pass@k Metric for RLVR: A Diagnostic Tool of Exploration, But Not an Objective," arXiv:2511.16231, 2025.
[67] P. Cordero-Encinar and A. B. Duncan, "Certified Self-Consistency: Statistical Guarantees and Test-Time Training for Reliable Reasoning in LLMs," arXiv:2510.17472, 2025.
[68] Q. Di, K. Ji, X. Li, H. Zhao, and Q. Gu, "Best-of-Majority: Minimax-Optimal Strategy for Pass@k Inference Scaling," in Proc. Int. Conf. Learn. Represent. (ICLR), 2026. arXiv:2510.03199.
[69] Z. Cen, Y.-F. Yao, W. J. Han, Z. Liu, and D. Zhao, "Behavior Injection: Preparing Language Models for Reinforcement Learning," arXiv:2505.18917, 2025.