Where RLOO Stops on Countdown: A Capability Ceiling at the Additive $\leftrightarrow$ Multiplicative Boundary

Aarohi Gupta aarohig@stanford.edu

Extended Abstract

Motivation and problem. Reinforcement learning on verifiable rewards is the main lever for getting reliable reasoning out of small language models cheaply, so where it stops improving them is a compute-allocation question. A standard concern about online, group-relative RL (RLOO (1), GRPO (4)) is *advantage collapse*: within-group rollouts converge, the group-relative advantage falls to zero, and learning stalls. We ask whether advantage collapse is what stalls RLOO on the Countdown arithmetic task with Qwen-2.5 0.5B, and if not, what does, and whether the standard RL toolkit can fix it.

Method and novelty. We run a controlled diagnostic study, treating the collapse hypothesis as a claim to be falsified rather than assumed. We instrument a baseline RLOO run with diagnostics beyond the standard optimisation metrics: structural (AST) rollout diversity, effective sample size, a three-way zero-advantage decomposition, and reward composition. The originally proposed intervention (a coupled operand-count curriculum with a temperature spike, designed before seeing data) becomes one hypothesis among several: after the baseline refutes collapse we add reward shaping, mode-triggered exploration, and rejection fine-tuning, two controls (a doubled generation budget; an operator-targeted fine-tune), and a per-prompt capability map backed by a brute-force structural audit. The novelty is the diagnosis, localising where and why small-model RLOO plateaus, not a new optimiser.

Implementation and headline results. Qwen-2.5 0.5B is SFT-warm-started and trained with RLOO (group size 8, 100 steps) on Countdown 3-to-4-operand problems; we report pass@k to 64 over two baseline seeds (a $\sim$5 pp noise floor). (i) The collapse hypothesis is half-refuted: structural diversity declines ($0.87 \rightarrow 0.59$) but the gradient signal does not vanish, the zero-advantage fraction holding flat at $\sim$28% while its composition flips from all-format-wrong to all-correct. (ii) An unanticipated failure takes its place, a no-answer regime climbing from 2% to 26% of rollouts, 97.7% of which exhaust the 1024-token budget mid-search rather than abstaining. (iii) No intervention or control improves baseline pass@1 (four worsen it by 5 to 9 pp, two are within noise). (iv) Across 50 prompts $\times$ 5 conditions $\times$ 64 samples, 22% of prompts are solved by no method at any budget, and all eleven require $\times$ or $\div$; where the model reaches for them it recognises single-step divisibility (e.g. $78/39 = 2$) but never chains it into the multi-step expression the solution needs.

Discussion, limitations, and conclusion. The pass@1 $\leftrightarrow$ pass@64 gap is best read as a capability boundary between additive and multiplicative composition rather than an optimisation failure the toolkit can close, a concrete, localised instance of RL sharpening the base model without expanding what it can reach (3). The main limitations are single-seed interventions (effects within the $\sim$5 pp floor are not claimed significant) and a single model and task (n=11 unreachable prompts, deterministically enumerated rather than sampled); a mid-project train/test leakage was handled by excluding the shared keys. Moving this ceiling needs levers outside the online-RL toolkit, such as model scale, compositionally targeted data, or inference-time search; an all-negative intervention table is itself informative about where RL fine-tuning stops paying off.

Abstract

Online RL on small reasoning models is often described as collapsing when within-group rollouts converge and group-relative advantages vanish. We instrument RLOO on Countdown 3-to-4-operand arithmetic with Qwen-2.5 0.5B to test that account and find a different one. Diversity does decline, but the gradient signal does not vanish; instead, a failure the collapse framing does not predict takes over, with a growing fraction of rollouts producing no answer and exhausting the generation budget mid-search. We attack this failure along every axis the RL toolkit offers (a difficulty curriculum, reward shaping, exploration, rejection fine-tuning, a doubled generation budget, and an operator-targeted fine-tune), and none improves single-sample accuracy. A per-prompt audit explains why: 22% of test prompts are solved by no method at any sampling budget, and these are precisely the problems whose solutions require multiplication or division. Where the model does reach for $\times$ or $\div$, it recognises single-step divisibility but does not chain it into the multi-step expression the solution needs. The pass@1 $\leftrightarrow$ pass@64 gap on this task looks less like an optimisation failure than a capability boundary between additive and multiplicative composition.

1 Introduction

Reinforcement learning on verifiable rewards is the main lever for getting reliable reasoning out of small language models cheaply, so where it stops improving them is a question about how to spend limited compute. We study this on Countdown, training Qwen-2.5 0.5B through SFT, IPO, and RLOO, where a target integer must be reached from three or four operands. A standard concern about online RL on tasks like this is advantage collapse: as training proceeds, within-group rollouts converge, the group-relative advantage estimator falls to zero, and learning stalls. Two mechanisms drive it: problems the policy cannot solve yet (all-incorrect groups) and rollouts that look identical (low within-group diversity). Our intervention couples both, a difficulty curriculum on operand count with a transient temperature spike at each advancement to restore exploration pressure when the policy meets harder problems.

This frames two questions: *(RQ1) is advantage collapse the mechanism that stalls RLOO on Countdown, and (RQ2) if not, what is, and can the standard RL toolkit fix it?* We treat the hypothesis as a claim to be tested rather than assumed, and instrument the baseline run to check it. The diagnostics falsify half of it: diversity declines, but the gradient signal does not vanish, and a failure the hypothesis never anticipated appears in its place. A large fraction of rollouts produce no parseable answer and exhaust the generation budget mid-search (Section 4.1). The rest of the paper follows that discovery. We attack the failure from four mechanism-distinct directions (data, reward, sampling, imitation) and two controls (a doubled generation budget; an operator-targeted fine-tune), and none recovers baseline pass@1; the investigation converges instead on a structural account of where the policy stops (Sections 4.3–4.6). Our contribution is the diagnosis: identifying *where* small-model RLOO plateaus on Countdown and *why* that plateau resists every intervention axis the standard RL toolkit offers.

2 Related work

Group-relative RL. RLOO (1) estimates a per-prompt baseline from the other rollouts in a group, avoiding a learned value network; GRPO (4) is the closely related group-relative method, and both share the failure we study. The group-relative advantage vanishes when a group's rollouts are scored identically, which is the starting point for the diversity-and-coverage concern we set out to test.

Countdown and search behaviour. Countdown is a standard probe for search-style reasoning in small models; Gandhi et al. (2) tie self-improvement on it to cognitive behaviours such as verification and backtracking. The malformed regime we find is the failure mode of that behaviour: a backtracking search that never commits before the budget runs out. Truncated, over-long RL rollouts are a documented pathology, from length exploitation in RLHF (7) to the overlong-response handling large-scale RL systems build in (8); our no-answer regime is the same surface phenomenon, but

here the length is genuine search exhausting the budget rather than reward-hacking, and it proves a symptom rather than a cause (Section 4.3).

Does RL expand capability? Yue et al. (3) argue that RL on verifiable rewards sharpens the base model toward answers it can already reach but does not enlarge the set reachable at large k, an aggregate effect measured across benchmarks. Our contribution is to localise that effect: we show *which* prompts form the unreachable set and *why*, a structural signature at the additive $\leftrightarrow$ multiplicative boundary that none of our six interventions and controls moves.

3 Methods

3.1 Experimental setup

The base model is Qwen-2.5 0.5B, SFT-warm-started on `Asap7772/cog_behav_all_strategies`. The RLOO training data is `asingh15/countdown_tasks_3to4`, with 490,314 train rows and a 50-row test set. The split is not fully disjoint: 12 train rows share $(\text{target}, \text{sorted}(\text{nums}))$ with 11 test rows, so every run here excludes those 11 keys from training, and the original leaky RLOO numbers are reported for context only. Each Countdown prompt provides a target integer and 3 or 4 operand integers; a correct response wraps an arithmetic expression using each operand exactly once in an `<answer>...</answer>` block that evaluates to the target. The reward function returns 1.0 for an arithmetically-correct answer, 0.1 for a parseable-but-wrong answer ("format-only"), and 0 for any rollout with no parseable answer block ("malformed"). RLOO uses group size $K{=}8$, batch size 128, learning rate 10^{-5}, KL coefficient 10^{-3} to the SFT reference, entropy coefficient 10^{-3}, training temperature 1.0, max generation tokens 1024, and 100 training steps. Evaluation uses temperature 0.6, top-k 20, top-p 0.95; pass@k is computed from 64 samples per prompt unless noted.

We compare against three reference points from the default pipeline: SFT (the warm-start floor), IPO (an offline-preference comparison), and the original RLOO run on the uncorrected split. We report pass@k out to $k{=}64$ in full rather than a single operating point, because the gap between pass@1 and pass@64 is the object of study. Two baseline seeds (0 and 1) set a noise floor of $\sim$5 pp on per-prompt accuracy; single-seed intervention effects within it are reported but not claimed significant.

3.2 Probe set

30 prompts are held out of training to support per-prompt longitudinal analysis. The set is stratified $2{\times}3$ over operand count $\in \{3, 4\}$ and target tier (the size of the target integer) $\in \{{<}40,\ 40\text{–}69,\ {\geq}70\}$ with five prompts per cell, drawn deterministically with seed 0 from the train split. Every condition uses the identical 30 prompts. At each checkpoint saved during training (every 10 steps), we sample $K{=}8$ rollouts per probe prompt via vLLM at the training-sampling settings.

3.3 Diagnostics

Standard RL training metrics (loss, pg_loss, kl_loss, mean entropy, mean importance weight, mean advantage, grad_norm, learning rate, mean reward) track optimisation health but do not distinguish the collapse hypothesis from its alternatives. We add the following diagnostics:

- **Structural diversity (answer AST).** Each rollout's final `<answer>` block is parsed to an AST and fingerprinted by operator-tree shape (commutative children sorted, so $93{+}61$ and $61{+}93$ match) and by the sorted multiset of intermediate values; the metric is the fraction of distinct fingerprints per group. Unlike edit distance, this is not fooled by token-level differences that encode the same reasoning path.
- **Effective sample size $K_{\text{eff}} = K(1 - \bar{s})$,** with $\bar{s}$ the mean within-group rollout similarity (mean pairwise normalised token-level edit distance, via `rapidfuzz`). It summarises how much of the $K{=}8$ rollout budget is non-redundant.
- **Zero-advantage decomposition.** When all K rollouts share a reward, the group-relative advantage is zero and contributes no gradient. We split these groups by the shared reward (all-correct / all-format-only / all-malformed): the collapse hypothesis predicts starvation specifically from the all-malformed case, distinct from the harmless "everyone already solves it" case.

- **Reward composition.** Per-step fraction of rollouts in each bucket $\{0, 0.1, 1\}$; the middle bucket isolates "learned to format, didn't learn to solve."

Each step's full batch is also serialised to JSON, so every rollout-derived metric in this paper (length signatures, qualitative inspection) is a post-hoc transformation computed without re-training.

3.4 Interventions and controls

The coupled curriculum was designed before the baseline run, as a direct test of the advantage-collapse hypothesis; the other three target the malformed regime the baseline reveals (Section 4.1). Figure 4 shows where each condition acts on the RLOO loop, and Appendix A gives the curriculum and temperature-controller pseudocode.

- **Coupled curriculum and temperature spike** (data + sampling). A state machine (5) gates training data by operand count, starting on 3-operand prompts and advancing (monotonically) to all prompts once the 5-step rolling-mean batch reward crosses 0.6; at advancement the sampling temperature spikes $1.0 \to 1.3$ and decays back with $\tau{=}10$.
- **Format shaping** (reward). The malformed-rollout reward drops $0.0 \to -0.1$; correct and format-only rewards are unchanged. Tests whether reward pressure redistributes mass away from the malformed bucket.
- **Mode-triggered exploration** (sampling). The same spike-and-decay, triggered instead when the rolling-3-step malformed fraction crosses 0.15 (cooldown 5); no data curriculum. Tests sampling pressure rather than reward pressure on the same target as format shaping.
- **Rejection fine-tuning** (imitation). All training rollouts are filtered to reward 1.0, deduplicated, and used for a brief SFT pass from the baseline's final checkpoint. Tests whether the pass@1/pass@64 gap is bridgeable by imitating the model's own successes.

Two further conditions are controls rather than candidate fixes:

- **Budget extension.** Re-evaluate the baseline at `max_tokens`$= 2048$, all else fixed. Tests whether the malformed regime and the failed prompts are a generation-budget artefact.
- **Multiplicative RFT.** The same RFT pipeline, but the filter also requires the answer to contain $\times$ or $\div$. Tests whether the operator prior is the binding constraint.

4 Results

4.1 Baseline dynamics and the regime split

Table 1 reports the scalar diagnostics; the reward-composition and zero-advantage trajectories are in Figure 1. Mean per-token entropy declines monotonically from 0.41 to 0.26 (a 36% relative drop), tracking the original (uninstrumented) RLOO run within sampling noise. The instrumentation is non-perturbative. Structural diversity falls from 0.87 to 0.59; K_{eff} from 5.85 to 5.33 of $K{=}8$. The diversity side of the original hypothesis is visible.

Table 1: **Baseline RLOO scalar diagnostics (seed 0).** Step-0 to step-99 deltas for the metrics not shown in Figure 1; the reward-composition and zero-advantage trajectories appear there, and the full table is Table 4. The two seeds agree to within 5 pp on every metric at step 99.

Metric	**Step** 0	**Step** 99	Δ
Reward mean	0.31	0.56	$+0.25$
Mean per-token entropy	0.41	0.26	-0.15
K_{eff} (of 8)	5.85	5.33	-0.52
Structural diversity (ops)	0.87	0.59	-0.29
KL to SFT reference	0.0	46.7	$+46.7$

What the hypothesis predicted but the data refutes is gradient starvation (Figure 1, bottom). The total zero-advantage fraction is at $\sim 28\%$ throughout training. It's composition changes: at step 0 it is dominated by all-format-wrong groups (27% of all groups, 96% of zero-adv groups); by step 99 it

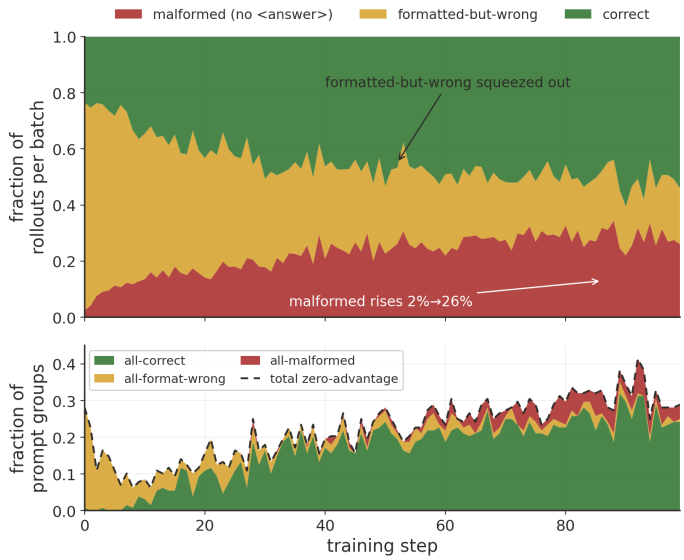

Figure 1: **The baseline refutes the gradient-starvation hypothesis and surfaces a different failure.** Top: per-step batch fraction in each reward bucket. The formatted-but-wrong middle (amber) collapses from 74% to 20% while malformed rollouts (red) rise from 2% to 26%. Bottom: the zero-advantage fraction decomposed into its three all-identical-reward sub-categories. The total (dashed) is flat at ∼28% throughout training, but its composition flips from all-format-wrong (amber) to all-correct (green); the hypothesised all-malformed gradient-starved mode (red) never exceeds 4%. The gradient signal recomposes instead of vanishing.

is dominated by all-correct groups (24% of all groups, 84% of zero-adv groups). The hypothesised all-malformed gradient-starved failure mode never exceeds 4% at any step. The model just learns to solve many prompts so consistently that within-group reward variance disappears for the easy ones.

The reward composition (Figure 1) shows a third phenomenon that was not part of the original framing. The fraction of malformed rollouts (no parseable `<answer>` block) rises monotonically from 2% to 26%, while the formatted-but-wrong middle collapses from 74% to 20%. RLOO partitions prompts into a clean bimodal regime (solved cleanly or producing no answer at all), and the arithmetically-wrong-but-formatted middle is squeezed out.

Truncation control on the malformed regime. We initially interpreted the malformed regime as a behavioural "I don't know" signal. A direct test refutes that reading. Tokenising every malformed rollout in the baseline run with the Qwen tokenizer (we sampled 2,735 across steps 0, 25, 50, 75, 99): mean token count is 1014.7 (p_{99}=1024), 97.7% **of malformed rollouts terminate within** 5 **tokens of the** 1024**-cap, and** 0% **end with the natural EOS token** `<|im_end|>`. For comparison, correct rollouts average 338 tokens and format-only rollouts average 546 tokens. Each bucket holds a stable length signature across training (Figure 5). Malformed rollouts are not deliberate non-commitments; they are the policy exhausting its generation budget on extended trial-and-error search. The qualitative pattern is visible in the rollout text: "$70 - 67 = 3$, no, try ... Final attempt: $67 + 3 = 70$, ..." with the rollout cut mid-clause when the cap fires.

4.2 Per-prompt longitudinal analysis on the probe set

The probe trajectories localise the malformed regime to the prompts the model cannot solve (Figure 8). At the final checkpoint, 3-operand probes reach 0.52 accuracy with 16% malformed, while 4-operand probes asymptote at 0.19 accuracy with 42% malformed: the no-answer mode is a 4-operand phenomenon, not a uniform property of training. The same trajectories expose clean negative transfer in the coupled curriculum (+5 pp on 3-operand probes, −6 pp on 4-operand): it specialises on the 3-operand prompts it spent stage 0 on and forgets the 4-operand prompts it saw only after the curriculum advanced at step 12.

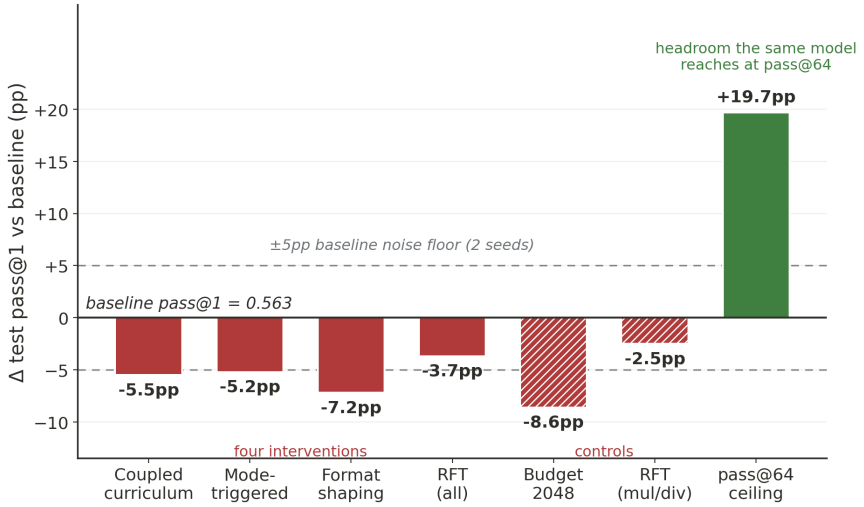

Figure 2: **No axis of the RLOO toolkit improves pass@1.** Change in test pass@1 relative to the seed-1 baseline (0.563); the zero line is baseline. None of the four interventions (solid) or two controls (hatched) rises above it: four fall 5.2–8.6 pp below, outside the ±5 pp two-seed noise floor (dashed), and two (RFT, mul/div-RFT) sit within noise of baseline. The green bar is the +19.7 pp of headroom the same checkpoint reaches at pass@64, the gap no condition recovers.

4.3 Intervention results: pass@k and mechanism

Table 2: **Pass@k on the 50-prompt test set.** Baseline seed 1 sets the headline reference (seed 0 matches it at pass@1, both 0.563). Evaluations at k=64 samples per prompt unless noted; SFT, IPO, and the original (pre-exclusion) RLOO run are at k=16. RFT and mul/div-RFT lie within the ±5 pp two-seed noise floor of baseline.

Condition	pass@1	pass@4	pass@8	pass@16	pass@32	pass@64
SFT (`k=16`)	0.281	0.593	0.708	0.760	—	—
IPO (`k=16`)	0.393	0.670	0.744	0.800	—	—
RLOO original (`k=16`)	0.558	0.690	0.708	0.720	—	—
RLOO baseline, seed 1	**0.563**	0.689	0.716	0.739	0.755	**0.760**
Coupled curriculum	0.508	0.668	0.696	0.716	0.729	0.740
Format shaping	0.491	0.643	0.671	0.690	0.705	0.720
Mode-triggered	0.511	0.670	0.702	0.724	0.740	**0.760**
RFT, all correct rollouts	0.526	0.670	0.697	0.716	0.729	0.740
Baseline @ `max_tokens`= 2048 (control)	0.477	0.658	0.684	0.700	0.718	0.740
RFT, mul/div correct only (control)	0.538	0.690	0.714	0.728	0.735	0.740

Table 2 and Figure 2 give the result up front: no intervention or control improves baseline pass@1. Four of the six fall 5.2 to 8.6 pp below it, beyond the ±5 pp two-seed noise floor; the other two (RFT and mul/div-RFT) are within that floor, indistinguishable from baseline rather than genuine regressions. Pass@64 never rises above 0.760. We read each condition by the problem it was meant to fix.

Coupled curriculum: the original hypothesis. If advantage collapse stalls learning, easing the policy into hard problems while re-injecting exploration should help. It does not: pass@1 falls 5.5 pp, and the probe trajectories (Section 4.2) show why, the curriculum specialising on 3-operand prompts and regressing on 4-operand ones.

Format shaping: penalise the no-answer regime. If the malformed mode is the problem, penalising it should redistribute mass toward answers. It zeroes the reward-0.0 bucket by construction

(malformed rollouts now score -0.1), and the reward-independent no-answer fraction does fall, from 26% to 14% (Figure 6). But the freed mass routes to formatted-but-wrong, not correct (frac_correct $0.54 \to 0.48$, format-only $0.20 \to 0.38$): forbidding the no-answer output makes the model commit more wrong answers, and pass@1 falls 7.2 pp, the largest drop of any condition.

Mode-triggered exploration: perturb the same regime by sampling. The same target as format shaping, attacked with sampling pressure instead of reward. The trigger fires 17 times; after the first at step 17 the cooldown floor binds, so the run trains at $T \approx 1.2$ from step 17 on rather than in transient bumps. Pass@1 falls 5.2 pp, yet this is the one condition that holds baseline pass@64 (0.760) despite the largest KL movement of the four ($46.7 \to 123.7$). On a single seed we read this as an observation rather than a robust effect: sustained sampling pressure appears to move the policy far in policy space without improving any single attempt, while preserving the set of prompts solvable somewhere in 64 samples, a possible trade-off of short-tail commitment for long-tail diversity rather than a pure negative.

Rejection fine-tuning: imitate the model's own successes. If the pass@1/pass@64 gap is just under-exploited competence, fine-tuning on the model's own correct rollouts should close it. It does not move the needle (-3.7 pp, the smallest drop): the imitation target is the additive successes the model already makes, which are not the bottleneck.

4.4 Generation-budget control

The natural follow-up to the truncation finding is to ask whether more generation budget would unblock the malformed rollouts. We re-eval the baseline checkpoint at `max_tokens`$= 2048$. Pass@1 falls by 8.6 pp ($0.563 \to 0.477$) and pass@64 falls by 2.0 pp; per-operand accuracy drops on both 3-op (-10.6 pp) and 4-op (-6.9 pp). Notably, 0 of 11 universally-failed prompts (identified in Section 4.5) are unblocked. The malformed-rollout fraction at the 2048-token budget remains 22% with mean token count 1890. Malformed rollouts do not extend to the new cap; they terminate at roughly the same effective budget the policy was trained to use.

The interpretation is a train/test mismatch on completion length: the policy was conditioned on a 1024-token budget, and the extra reasoning the larger budget allows produces wrong commitments more often than correct ones. The control rules out budget as the locus of the ceiling, the first evidence that more *compute per attempt* does not help.

4.5 Per-prompt capability map

Table 3: **Capability map across** 50 **test prompts,** 5 **conditions** $\times$ 64 **samples each.** Universally solved $\equiv$ every condition ≥ 0.5 success; universally failed $\equiv$ every condition $= 0.0$.

Category	N	%	3-op	4-op
Universally solved	23	46%	17/24 (71%)	6/26 (23%)
Variable	14	28%	5/24 (21%)	9/26 (35%)
Universally rare ($< 5\%$)	2	4%	1/24 (4%)	1/26 (4%)
Universally failed	**11**	**22%**	1/24 (4%)	**10/26 (38%)**

Five conditions $\times$ 64 samples each is 320 attempts per test prompt. We compute the mean success fraction across these attempts per prompt. The partition is bimodal (Table 3; Figures 9 and 7 show per-prompt heatmap and distribution): 46% of prompts are universally solved ($\geq 50\%$ success by every condition) and 22% are universally failed (0 correct in any of the 320 attempts). The latter are almost all 4-operand (10/11) and form the operational ceiling for our study: the pass@64 saturation at $\sim$0.76 corresponds to the $\sim 26\%$ of prompts no method ever solves.

Structural audit of the universally-failed subset. For each problem we enumerate all valid arithmetic expressions over the operand set using brute force (each 3-op problem yields $\sim$200 candidates, each 4-op problem $\sim$8000). We then label each problem by whether *any* solution avoids $\times$ and $\div$. The boundary is sharp: 11 **of** 11 **universally-failed problems have a simplest solution that requires** $\times$ **or** $\div$**, and** 0 **of** 11 **admit any addition/subtraction-only solution.** This

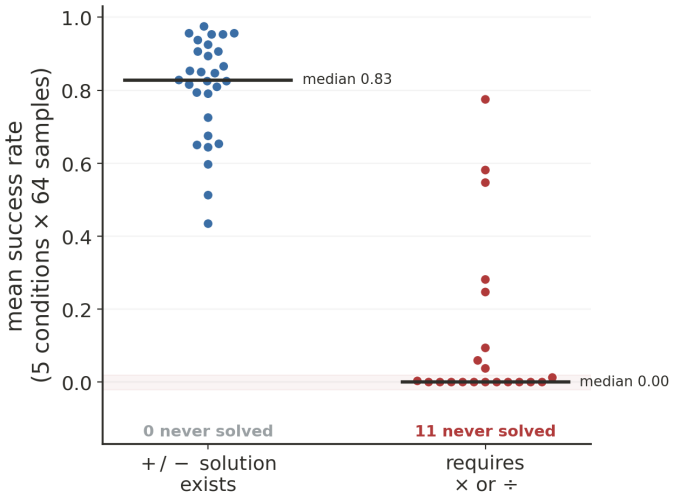

Figure 3: **The capability ceiling is the additive ↔ multiplicative boundary.** Mean success rate per test prompt (5 conditions × 64 samples each), grouped by whether the operand set admits any addition/subtraction-only solution. Prompts with an add/sub path are solved reliably (median 0.83, none never-solved); prompts that require × or ÷ collapse to the floor (median 0.00), and all 11 prompts that no method ever solves fall in this group. The solution-density view of the same split is in Figure 10.

is a deterministic property of each problem rather than a sampled estimate: the full solution set is enumerated, so the $11/11$ is exact. By contrast, 22 of 23 universally-solved problems admit an add/sub-only solution. The single 3-op outlier in the failed set, target=44 with nums=$[2, 4, 26]$, is solvable only as $2 \times (26 - 4)$ and its commutative permutation; the maximum additive value reachable from $\{2, 4, 26\}$ is 32, so no additive expression even comes within 11 of the target. The capability ceiling has a single structural signature (Figure 3): the add/sub ↔ mul/div boundary.

Within the 4-operand subset, failed problems have on average 8 valid solutions, versus 111 for universally-solved 4-op problems (14× density gap; Figure 10). The model not only lacks the multiplicative insight; even exploratory sampling has many fewer targets to hit.

4.6 Operator-selection refinement

The capability signature suggests a sharp follow-up: does the model use × and ÷ at all, or are they absent from its repertoire? We measure the fraction of correct answers using × or ÷ across all 16,000 eval responses (5 conditions × 50 prompts × 64 samples). The model uses × or ÷ in 11.2% of correct answers under the baseline, with the rate ranging from 7.3% (format-shaping) to 12.2% (mode-triggered) across conditions. On the 11 universally-failed prompts specifically, where × or ÷ is structurally required, the attempt-rate is 1.7% to 4.3% depending on condition. A 10× gap between global usage and need-rate is present in every condition.

We probed whether the model's operator prior is the binding constraint by filtering RFT data to correct rollouts using × or ÷ (4,012 pairs after deduplication), then SFT-training on this subset from the baseline RLOO checkpoint. The intervention shifts the global ×/÷ rate from 11.2% to 13.3% (+2.1 pp) and the rate on failed prompts from 3.1% to 3.7% (+0.6 pp). Pass@1 moves from 0.563 to 0.538 (−2.5 pp, within the noise floor). **Zero of** 11 **universally-failed prompts get unblocked.**

The qualitative pattern in the rare ×/÷ attempts on the failed problems is informative. We collected every parseable rollout from the 6 conditions on those 11 prompts (∼4,000 attempts) and inspected the ×/÷ subset. The clearest pattern follows; two more (the simplest mul-only problem, and orbital additive attempts) are in Appendix D.

Single-step divisibility recognised; multi-step composition not. On target=82, nums=$[1, 39, 78, 83]$, whose known solution is $83 - ((78/39) - 1) = 82$: the model's ×/÷

attempts include $(83-1)-(78/39)=80$ and $(83-1)+(78/39)=84$. It correctly identifies $78/39=2$ and uses the $83-1=82$ sub-expression, but never composes them with the additional -1 the solution requires. On target=15, nums=$[28, 32, 39, 96]$ (solution $39-(96/(32-28))=15$), the model attempts $(96/32)-28+39=14$: it spots an integer division ($96/32=3$) but misses that the actual divisor must be *constructed* as $32-28$. The model has one-step divisibility insight; it does not have the two-step "subtract to construct a divisor" composition.

The operator-targeted RFT result, combined with the qualitative pattern, refines the bottleneck claim. The model has $\times/\div$ in its toolkit; the imitation training can shift its prior, but the failure may be in recognising the *structural cues* (divisibility-by-difference, subtract-to-construct-divisor) that route an additive trial-and-error process into the multiplicative search space the unsolved prompts require.

5 Discussion

5.1 Why convergence across mechanism axes matters

The six interventions and controls span data, reward, sampling, imitation, the generation budget, and the operator prior, and no single-axis hypothesis explains their joint failure. A wrong difficulty schedule does not explain format shaping or RFT; a mis-specified malformed reward does not explain mode-triggered; too small a generation budget predicts the budget extension to help, and it hurts instead. The only locus upstream of all six is the model's capacity to recognise the structural cues that route problem-solving into the multiplicative search space.

5.2 Refined claim and what would actually move the ceiling

The strongest version of the claim our data supports: the pass@1 $\leftrightarrow$ pass@64 gap of 0.20 on Countdown 3-to-4-operand with Qwen-2.5 0.5B is a capability ceiling at the *structural composition* of multiplicative operations, not the absence of $\times$ or $\div$ in the policy's repertoire, and not the training-time signal allocation that controls how often $\times$ or $\div$ gets reinforced. Three intervention classes are correctly motivated by this diagnosis and lie outside the project's compute and dataset constraints. **Model scale**: structural-composition pattern matching is the kind of capability that improves with parameter count on similar benchmarks. **Targeted training data**: Countdown variants specifically curated to teach divisibility-by-difference and similar two-step compositional cues. **Inference-time search procedures**: tree-of-thoughts decoding (6) or beam search over the operator space, which decouple the model's per-step token distribution from the structural search problem. We do not test these here; they are the natural follow-up directions consistent with the diagnosis.

5.3 Practical implications

An all-negative intervention table is itself informative. On this task, the plateau looks like a capability boundary rather than an optimisation bug, and if that pattern recurs elsewhere it would carry an allocation lesson: once the unsolved set is structural, further curriculum, reward, or sampling tuning sharpens pass@1 without moving the reachable frontier, so compute may be better spent on scale, targeted data, or inference-time search. We do not claim this generalises beyond Countdown 3-to-4 at this scale; what our setting offers is a cheap test of the distinction, namely whether the unsolved prompts share a structural property the policy cannot compose. Put plainly, an RL budget spent against a structurally homogeneous unsolved set is largely wasted: the gains accrue to prompts already within reach.

6 Limitations

Single seed per intervention. We run a single seed for each intervention, justified by the LLM-evaluation convention and the course guidance. The two-seed baseline (seed 0, seed 1) gives a noise floor of ~ 5 pp on per-prompt accuracy, which bounds the smallest effects we can claim. Intervention effects within 5 pp of baseline are reported but not asserted as significant.

Dataset leakage. Because the original RLOO run trained on the leaky split (Section 3.1) while our re-runs excluded the 11 shared keys, the comparison of the original 0.558 pass@1 to our instrumented 0.563 is approximate; we use the cleaner re-run as the headline reference.

Single model and dataset. All findings are on Qwen-2.5 0.5B and Countdown 3-to-4. The unreachable set is small (11 prompts), but each problem's operator requirement is a deterministic enumeration rather than a statistical estimate, so the open question is external validity, whether the additive $\leftrightarrow$ multiplicative ceiling recurs at 1.5B, 7B, or on Countdown variants with deeper chains, not whether $11/11$ is noise.

Probe granularity, eval temperature, and K_{eff}. The 30-prompt probe set has 5 prompts per cell, so per-cell variance is loose and we lean on the per-operand aggregate. Evaluation runs at temperature 0.6 against a training temperature of 1.0, standard but worth noting for the mode-triggered condition, which was trained hot. K_{eff} uses the linear $K(1-\bar{s})$; an inverse-similarity form gives qualitatively the same trajectories.

7 Conclusion

The pre-data hypothesis (advantage collapse stalls RLOO on Countdown) was refuted by the baseline: the gradient signal recomposes rather than vanishing, and the salient failure is a budget-exhausted no-answer regime the framing never anticipated. Six interventions and controls spanning every axis of the RL toolkit each fail to improve baseline pass@1, while 22% of test prompts go unsolved at k=64, every one of them requiring $\times$ or $\div$. The pattern points to structural composition of multiplicative operations, rather than their execution, as the binding constraint: the model recognises single-step divisibility but cannot chain it into the multi-step search the unsolved prompts need. Whether the ceiling moves with model scale, targeted curriculum data, or inference-time search is the natural follow-up.

Changes from the proposal

The proposal framed RLOO's plateau on Countdown as advantage collapse, with the coupled difficulty curriculum and temperature spike as the contribution and the expected fix. The baseline refuted that hypothesis: the gradient signal recomposes rather than vanishing, and instead a budget-exhausted no-answer regime emerged. The work consequently pivoted from proposing and validating a fix for advantage collapse to diagnosing what limits the policy. The coupled curriculum became one falsified hypothesis among several, alongside the reward-shaping, mode-triggered, RFT, budget, and operator-targeted conditions, and the per-prompt capability map and structural audit became the core; the out-of-difficulty (5-operand) generalisation probe in the proposal was cut for adding a distribution-shift confound. The headline contribution shifted from a method to a diagnosis.

References

[1] Ahmadian, A., Cremer, C., Gallé, M., et al. *Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs*. ACL, 2024.

[2] Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., Goodman, N. D. *Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs*. arXiv preprint, 2025.

[3] Yue, Y., Chen, Z., Lu, R., et al. *Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?* NeurIPS 2025.

[4] Shao, Z., Wang, P., Zhu, Q., et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. arXiv preprint, 2024.

[5] Bengio, Y., Louradour, J., Collobert, R., Weston, J. *Curriculum Learning*. ICML, 2009.

[6] Yao, S., Yu, D., Zhao, J., et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. NeurIPS, 2023.

[7] Singhal, P., Goyal, T., Xu, J., Durrett, G. *A Long Way to Go: Investigating Length Correlations in RLHF*. COLM, 2024.

[8] Yu, Q., et al. *DAPO: An Open-Source LLM Reinforcement Learning System at Scale*. NeurIPS 2025.

A Implementation details

All runs are seeded for `torch`, `numpy`, `random`, the training DataLoader, and per-step vLLM sampling, so trajectories reproduce across re-runs. The coupled curriculum and mode-triggered exploration share one temperature spike-and-decay controller, queried once per training step; the curriculum additionally gates the prompt pool.

```
# state, queried each of the 100 training steps
stage, spike_step = 0, None      # stage 0: 3-operand only;  1: all prompts

for step in range(100):
    r = rolling_mean(batch_reward, window=5)

    # coupled curriculum: monotone advance once reward clears 0.6
    if stage == 0 and step >= 5 and r > 0.6:
        stage, spike_step = 1, step

    # mode-triggered: (re)arm the spike on the malformed fraction
    # used in place of the curriculum trigger in that condition
    if rolling_mean(malformed_frac, window=3) > 0.15 and cooled_down(step):
        spike_step = step

    prompts = data if stage == 1 else only_3_operand(data)
    T = 1.0 if spike_step is None else 1.0 + 0.3 * exp(-(step - spike_step)/10)

    rollouts = sample(policy, prompts, temperature=T, K=8)
    rloo_update(policy, rollouts)
```

B Supporting figures

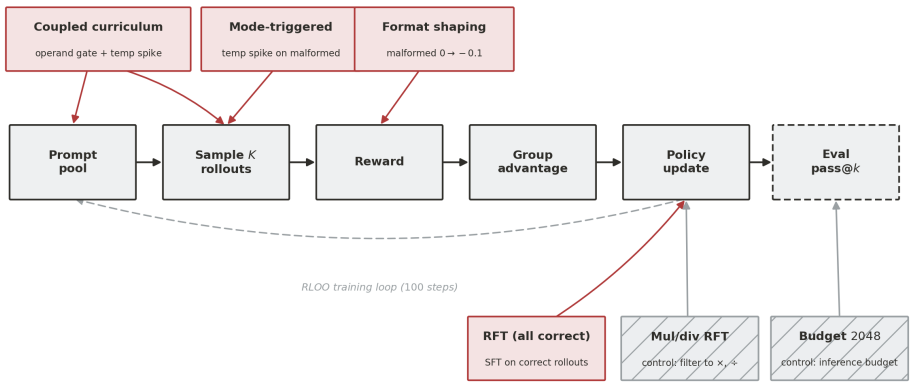

Figure 4: **Where each condition acts on the RLOO loop.** The five-stage training loop (prompt selection, rollout sampling, reward, group-relative advantage, policy update), with evaluation off-loop. The four interventions (red) and two controls (hatched) each perturb a different stage: the coupled curriculum gates the prompt pool and spikes the sampling temperature, mode-triggered exploration spikes temperature on the malformed signal, format shaping rewrites the malformed reward, and rejection fine-tuning imitates correct rollouts off-loop; the controls vary the operator filter and the inference budget. None modifies the group-relative advantage estimator or the model's capacity, where the ceiling sits.

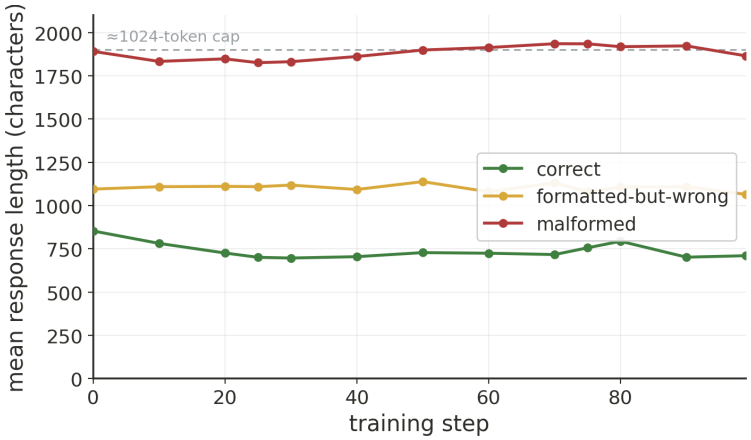

Figure 5: **Reward buckets are categorical behavioural modes with stable length signatures.** Mean response length (characters) per reward bucket over baseline training. The three modes hold their separation throughout: correct $\sim$700, formatted-but-wrong $\sim$1100, malformed $\sim$1900 ($\approx$1024 tokens, the generation cap). The malformed mean varies only between 1825 and 1934 while its population fraction rises $10\times$ (Figure 1): the regime split redistributes mass across fixed modes rather than lengthening rollouts within one.

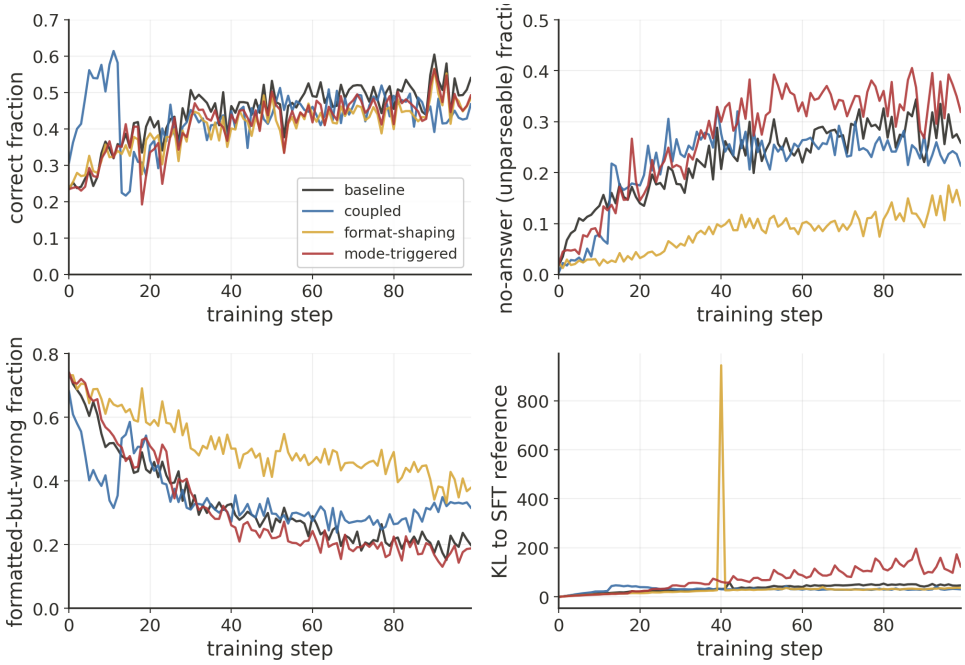

Figure 6: **Per-condition training trajectories.** The four conditions on correct fraction, no-answer fraction (1$-$parseable, measured reward-independently so format-shaping is comparable), formatted-but-wrong fraction, and KL to the SFT reference. Format-shaping (amber) is the only condition that suppresses the no-answer mode (to $\sim$0.14, not to zero), and its formatted-but-wrong fraction stays high: the penalised mass commits wrong answers rather than correct ones. Mode-triggered (red) raises both the no-answer fraction and KL, consistent with the sustained temperature elevation described in Section 4.3.

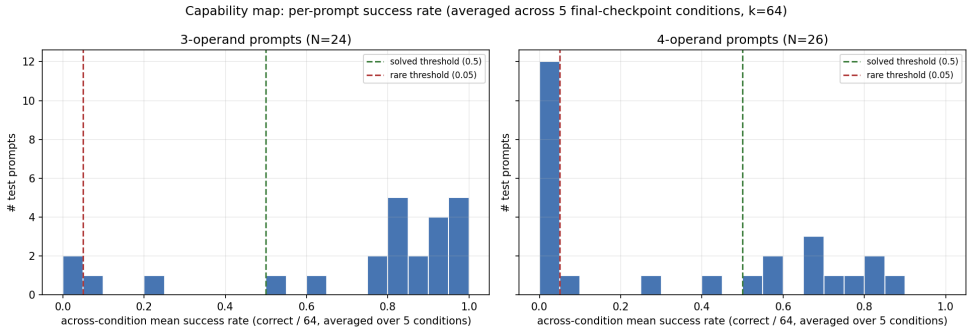

Figure 7: **The per-prompt capability distribution is bimodal.** Histogram of per-prompt mean success rate (across the 5 final-checkpoint conditions, 64 samples each), split by operand count. Three-operand prompts pile up near 1.0; four-operand prompts split into a solved cluster and a spike at 0 (12 prompts with <0.05 mean success, ten of them never solved), the quantitative form of the wall in Figure 9.

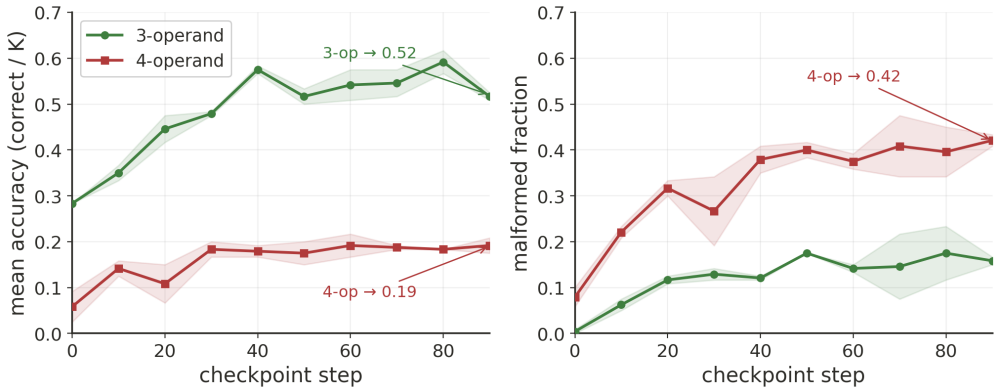

Figure 8: **The malformed regime is concentrated on the prompts the model cannot solve.** Baseline probe trajectories (mean over both seeds; shaded band spans the two seeds), averaged across the 15 three-operand and 15 four-operand probe prompts. Left: accuracy (mean correct out of $K=8$) per checkpoint: three-operand probes converge to 0.52, four-operand probes asymptote at 0.19. Right: malformed fraction, which rises to 0.42 on four-operand probes while staying near 0.16 on three-operand probes.

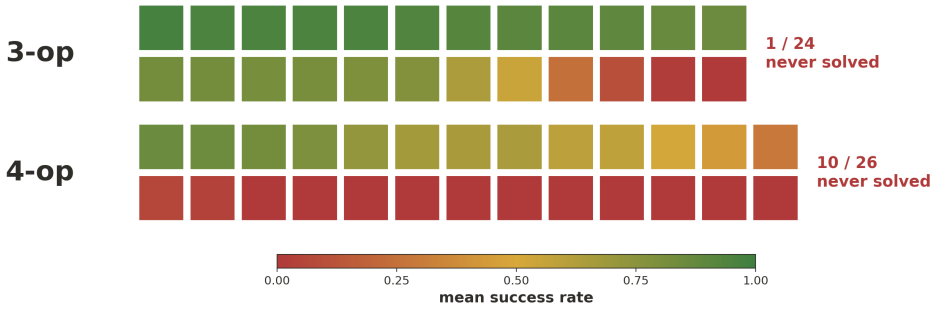

Figure 9: **Per-prompt capability map.** One square per test prompt, sorted within operand-count group by mean success rate across the 5 final-checkpoint conditions (baseline, coupled, format-shaping, mode-triggered, RFT) at 64 samples each. Each square's colour is the prompt's mean success rate. Red squares on the right of each row are prompts no condition solves at $k=64$: $1/24$ for three-operand prompts, $10/26$ for four-operand.

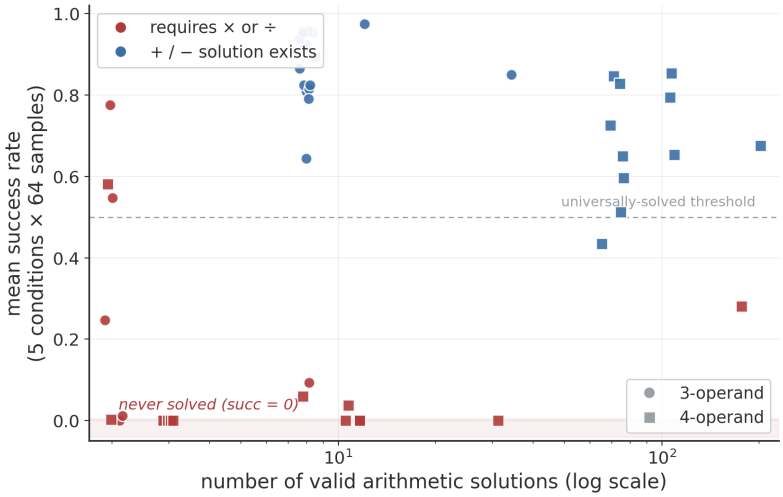

Figure 10: **Solution-density view of the capability ceiling.** Each point is a test prompt: mean success rate (5 conditions $\times$ 64 samples) against the number of valid arithmetic solutions the operand set admits (log scale), coloured by whether the simplest solution requires $\times$ or $\div$. The never-solved prompts are not only multiplicative but also solution-sparse (8 valid solutions on average versus 111 for solved 4-operand prompts), so they sit in a single low-density, multiplicative corner.

C Full baseline trajectory

Table 4: **Baseline RLOO trajectory, all diagnostics (seed 0).** Step-0 to step-99 deltas. The main-text Table 1 keeps the scalar rows; the reward-composition and zero-advantage rows here are plotted in Figure 1.

Metric	**Step** 0	**Step** 99	Δ
Reward mean	0.31	0.56	+0.25
Fraction correct	0.23	0.54	+0.31
Fraction format-only (wrong)	0.74	0.20	−0.54
Fraction malformed	**0.02**	**0.26**	**+0.24**
Mean per-token entropy	0.41	0.26	−0.15
K_{eff} (of 8)	5.85	5.33	−0.52
Structural diversity (ops)	0.87	0.59	−0.29
Total zero-advantage fraction	0.28	0.29	+0.01
zero-adv all-correct	0.01	0.24	+0.23
zero-adv all-format-wrong	0.27	0.01	−0.27
KL to SFT reference	0.0	46.7	+46.7

D Qualitative failure patterns

Two further patterns from the $\times/\div$ attempts on the universally-failed prompts (Section 4.6).

On the simplest mul-only problem, the model never samples $\times$. On target=44 with nums=$[2, 4, 26]$, a 3-operand problem whose only solutions are $2 \times (26 - 4)$ and its commutative permutation, we have 89 parseable attempts across 6 conditions $\times$ 64 samples. **Zero** use $\times$ or $\div$. The model's output orbits $(26 + 4) + 2 = 32$, the maximum additive value of the operand set, off by 12. Multiplication is physically required and never even attempted.

Additive attempts are orbital. On target=98, nums=$[20, 21, 31, 67]$, the model produces 44 valid-operand additive attempts at k=64 per condition. Five distinct expressions appear, all rebrackings of $67 + 31 \pm 20 \pm 21$, oscillating between 97 and 99. The model finds one additive arrangement near the target and re-permutes it rather than searching for structurally different paths.